FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Introduction to Structural Optimization using the ESO and BESO Evolutionary Methods

Filipe Assis Gonçalves

Master in Computational Mechanics

Supervisor at FEUP: Prof. Dr. Marco Parente

October 5, 2018

Introduction to Structural Optimization using the ESO and BESO Evolutionary Methods

Filipe Assis Gonçalves

Master in Computational Mechanics

October 5, 2018

Resumo

A otimização topológica é uma importante ferramenta para a engenharia mecânica. Atualmente com o aumento da busca por melhores designs, materiais mais leves, com maior resistência e menor custo, o uso de softwares de otimização é indispensável. A utilização desse tipo de software no processo de desenvolvimento de um produto também aumenta com computadores mais avançados e algoritmos mais eficientes. Este trabalho foi focado na implementação do algoritmo de Otimização Estrutural Evolutiva (ESO) e Otimização Estrutural Evolutiva Bidirecional (BESO) no software SOLIDFEM. Quatro algoritmos foram implementados, dos quais, três utilizaram o método ESO e um o BESO. Estudos foram conduzidos para identificar problemas como: padrão quadriculado e dependência da malha. Dois esquemas de filtragem, filtro elemental e nodal, foram utilizados com os algoritmos. Testes de sensibilidades foram realizados para definir o valor ótimo para parâmetros específicos, aplicabilidade em modelos 3D e eficiência dos algoritmos. Foi observado que todos os métodos tiveram bons resultados. No entanto, os métodos baseados em ESO tiveram resultados melhores. O método BESO apresentou grande potencial para melhorarias e usabilidade.

Abstract

Topology optimization is an important tool for mechanical engineering. Nowadays with the growing search for better designs, lightweight materials, with greater strength and lower cost the use of optimization software became necessary. The use of this type of software in the design process also increases with more advanced computers and more efficient algorithms. This work was focused on the implementation of the Evolutionary Structural Optimization (ESO) and Bi-directional Evolutionary Structural Optimization (BESO) on SOLIDFEM software. Four algorithms were implemented, three using ESO and one using the BESO method. Studies were conducted to identify checkerboard pattern and mesh dependency problems. Two filter schemes, elemental and nodal filters were used with the algorithms. Sensibility tests were conducted to define the optimum value of specific parameters, applicability with 3D models and the efficiency of the algorithms. It was observed that all methods had good results; however, the ones based on ESO had better results. The BESO method presented great potential to improve and usability.

Agradecimentos

Primeiramente agradeço a minha namorada, Alessandra, por ter sido minha companheira e incentivadora durante a jornada do mestrado. Pela compreensão principalmente durante a confecção da tese. Sem sua paciência para suportar as horas extras dispensadas na FEUP ou seu estimulo e carinho quando em vários momentos pensei não ser capaz de chegar até aqui. Obrigado meu amor!
Agradeço imensamente ao meu maior apoiador, tio Fernando. Muito obrigado pelo suporte que sempre deste-me, desde a minha infância até hoje. Não teria feito nem metade do que fiz ou teria chegado onde cheguei se não fosse por ti e meu avô Serafim (*in memorian*). O meu muitíssimo obrigado.
Aos meus pais, Paulo e Rosângela, pelo apoio ao longo desses anos. Todo o esforço que fizeram para que eu tivesse melhores oportunidades de estudo e condições para focar nos estudos. Muito Obrigado.
A minha avó Zenilda, por seu amor incondicional, seu carinho e conselhos que fizeram a distância ser menor do que realmente é. Obrigado Dona Zizilde.
Ao Henrique Cestaro, por todas as conversas que tivemos ao longos desses dois anos de curso, as horas de estudo em conjunto, as ajudas com programação, revisão de texto, por ter me mostrado novas músicas para estudar. Obrigado pela amizade. Aos meus amigos Raphael Scudiere, Sergio Griffo, Rubes Akira e Marcelo Moreno pela alegria diaria e incentivos mesmo estando em continentes diferentes.
Ao meu orientador, Prof. Dr. Marco Parente, por ter aceitado-me como orientando já com prazo reduzido, pela paciencia quando estive no início desse projeto e não conhecia a linguagem em Fortran ou como o SOLIDFEM funcionava. Muito obrigado!
Muito obrigado a todos vocês!

Filipe Assis Gonçalves

Because maybe, you're gonna be the one
that saves me.
And after all, you're my wonderwall

Oasis

Contents

List of Figures

List of Tables

Abbreviations and Symbols

Symbols

ΔL	[-]	Length variation
γ	[-]	Shear strain
λ	[-]	Lamé coefficient
ν	[-]	Poisson's coefficient
∇	[-]	Gradient
σ	$[N/mm^2]$	Stress
ε	[-]	Strain
τ	$[N/mm^2]$	Shear stress
A_0	$[mm^2]$	Cross-sectional area
c	[-]	Constitutive matrix
C	[-]	Compliance matrix
E	$[N/mm^2]$	Young's modulus
F	$[N]$	Force
G	[-]	Modulus of rigidity
H	[mm]	Hight
L_0	[N]	Load
K	[-]	Stiffness matrix
t	[mm]	Thickness
W	[mm]	Width

Abbreviations

α_{add}	Alpha addition variable
$\alpha^{th}_{adition}$	Alpha addition threshold variable
α_{adj}	Alpha evolutionary parameter
α_{del}	Alpha deletion variable
α^{th}_{delete}	Alpha deletion threshold variable
α_{del_max}	Maximum Alpha deletion variable
$\alpha_{element}$	Ordered Sensitivity Number Vector
AR	Addition Ratio
AR_{max}	Maximum Volume Addition Ratio
BESO	Bi-directional Evolutionary Structural Topology Optimization
BS	Beam structure

CP	Checkerboard Pattern
ESO	Evolutionary Structural Topology Optimization
EBSO	ESO Based Stress or Displacement Optimization
EBSD	ESO Based Stiffness or Displacement Optimization
EF	Elemental Filter
ELDELMAX	Maximum Alpha deletion variable
ER	Evolutionary Rate
ERR	Element Removal Ratio
ESN	Elemental Sensitivity Number
ESED	Elemental Strain Energy Density
EVR	Evolutionary Volume Ratio
FEA	Finite Element Analysis
FEM	Finite Element Methods
I-ADD-BESO	BESO Maximum Addition Index
IR	Inclusion Ratio
ISN	Improved Sensitivity Number
ISO	Isometric
NF	Nodal Filter
MD	Mesh dependency
MTS	Michell type structure
OP	Optimization Process
PI	Performance Index
PI-NORM	Performance Index Normalized
RR	Rejection Ratio
RR_i	Rejection Ratio for the iteration i
SC	Short cantilever
SN	Sensitivity Number
TBF	Two bar frame

Chapter 1

Introduction to Structural Optimization

This work is focused on the Structural Mechanical Engineering field. Therefore, two definitions need to be set first: *structure* in mechanics can be defined as "any assemblage of materials which is intended to sustain loads." according to J.E.Gordon [Gordon and Stewart, 1980]. *Optimization* means "making things the best", Christensen and Anders [2008]. By analogy *structural optimization* can be understood as "the goal of making an assemblage of material that sustains loads in the best way", Christensen and Anders [2008]. Another definition needed is the meaning of the word *best*[1], that denotes "of the most excellent or desirable type or quality" which applied to optimization can be interpreted as making the structure as stiff as possible, or as light as possible.

To reach the optimum design, some objectives need to be considered, like: minimize the weight, the volume, stress, vibration, cost, maximize the stiffness, search for homogeneous stress distribution, manufacturability, and prevent buckling and, or instability. These objectives are sometimes contradictories, more than one can occur at the same time in a project such as minimizing the mass and maximize the stiffness of a critical part. Another topic that is essential to optimization is the constraints. Independently if the goal is to maximize or minimize an objective and how many goals the model have, the use of constraints is necessary to apply the process correctly. Some examples of parameters that may be constrained are stress, displacement, and geometry Christensen and Anders [2008].

However, to accomplish it, independently the number of objectives involved and constraints applied, there is a crucial question: "how to decide where to place/remove material in a design domain previously defined to obtain the best structural performance?" Christensen and Anders [2008]. To answer this questions, the use of optimization methods is necessary.

1.1 The design process

The main steps of the design of a product are shown below Kirsch [1993]:

1. Function: what is the purpose of the product? Where will it be used? Who will use it and how?

[1] Definition from https://en.oxforddictionaries.com/definition/best

2. Conceptual design: what type of concept should be used? Which concept met the functional requirements?

3. Optimization: Knowing the concept and the constraints inherent to the function proceed to search for the best design possible

4. Details: Step controlled by social, market or aesthetic factor

Thinking about these steps, we can use a car as a product example. The primary function is transport people and, or products. In this case, the number of passengers that can be transported or the maximum cargo is vital for it. On the second step, the conceptual design, the question is about the size of the car, the number of doors, type of engineer, type of fuel, ... The third step, optimization, search for minimizing the cost, fuel consumption, and maximize the number of passengers or cargo. On the last one, the details, one may choose the color or the flooring material of the car. Paying particular attention to step 3, one way to realize it is an *iterative-intuitive method*. It is described as presented below:

a) Initial design

b) Investigation of the function specifications

c) If the specifications are not satisfied, a new design is proposed

d) With the new design, the process returns to step b)

e) If the specifications are satisfied, the process is stopped

Step 3 and b applied to mechanical structures uses computer-based methods such as finite element methods (FEM). It means that every iteration will be analyzed with accuracy. FEM will be presented and discussed in Chapter 5.

1.2 General mathematical form of a structural optimization problem

The next definitions are always present in optimization problems:

- Objective function (f): this function has the purpose to establish a rank of designs. f generates a grade which classifies whether the result is a good or bad design. Used to analyze designs measuring weight, displacement in a specific direction, effective stress or even cost of production

- Design variable (x): it can be a function or a vector that describes the design. It can be changed during the optimization process. It can describe the geometry or the materials applied in the domain.

- State variable (y): it can be a function or vector that represents the response of the structure, given the design x. The response can be displacement, stress, strain or force.

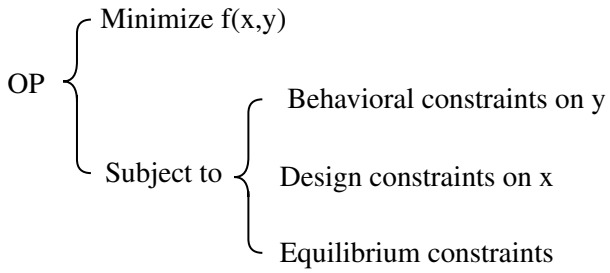


Figure 1.1: Generic process of optimization

The optimization process (OP) can be explained in a generic structural case as presented in the Figure 1.1:

There are three types of structural optimization problems: sizing, shape, and topology.

Sizing optimization: Refers to cases where the x is a type of structural thickness. Could be the cross-section areas of truss members or thickness of a plate.

Shape optimization: This is when x represents the contour or the form or the boundary of the structure, the shape of the contour changes, the equation that rules it does not change, in an optimal way. Here the connectivity of the structure does not change, and new boundaries are not formed.

Topology optimization: The most general form of structural optimization in a truss case, discrete, it takes the cross-section of the bars as design variables and allowing then to take value zero. It means that bars are removed from the structure. Here the connectivity is variable. When in a continuum type case, a two-dimensional sheet, the thickness of the sheet can be turned to zero. In a three-dimensional case, the density is the variable. In this case, the variable only can take the value 1 or 0.

Chapter 2

Evolutionary Structural Topology Optimization - ESO

2.1 Introduction

The Evolutionary Structural Topology Optimization (ESO) method has been developed since the early 1990s when the method was presented by Xie and Steven for the first time [Xiet and Steven, 1992]. The ESO concept is based on gradually removing inefficient material form a structure. The ultimate structure is the result of the evolution of itself along the process [Huang and Xie, 2010b]. This technique provides a powerful tool for engineers or anyone who searches for improving the forms and shapes of a model during the initial stage of a project. Some characteristics and deficiencies of the method will be present in this chapter.

2.2 ESO Based Stress or Displacement Optimization - EBSO

Monitoring the stress of the structure is the easiest way to detect if there is any part that is overload or unloads. Also, it is pertinent to say that, a good design has equilibrated stress values along the structure. So, using tools like finite elements analysis as FEA a stress map of the structure can be created, and a rejection criterion based on the stress level. When the element has a low-stress level, it can be deleted, since it has not been appropriately used. On the other hand, if it has a high-stress level, it is secure to say that more elements are needed in that area.

To determine if one element needs to be deleted, the system uses the von Mises stress, where the stress of one element, $\sigma_e^{\upsilon m}$, is compared with the maximum stress of the whole structure $\sigma_{max}^{\upsilon m}$. If the ratio betwen the $\sigma_e^{\upsilon m}$ and $\sigma_{max}^{\upsilon m}$ is under a rejection ratio RR_i previously defined for iteration i, the element is eliminated from the model.

$$\frac{\sigma_e^{\upsilon m}}{\sigma_{max}^{\upsilon m}} > RR_i \tag{2.1}$$

where RR_i is the current rejection ratio (RR). The RR_i is used along the optimization process (OP) up to the steady state is reached, when there are no more elements to be deleted. When this happens, the evolutionary rate (ER) is added in order to upgrade the RR_i value, (2.2), and the evolutionary process continues.

$$RR_{i+1} = RR_i + ER \tag{2.2}$$

2.3 ESO based Stiffness or Displacement Optimization - EBSD

Another way to improve a structure is by making use of its stiffness. The inverse of the compliance matrix, **C**, also known as stiffness matrix, **K**, is considered in this case. The **C** can be calculated using the (2.3). Where **f** is the force vector and **u** is the displacement vector.

$$\mathbf{C} = \frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{u} \tag{2.3}$$

Remembering that the inverse of (2.3):

$$\mathbf{K}\mathbf{u} = \mathbf{f} \tag{2.4}$$

Every iteration in which an element is removed generates a new stiffness matrix, as

$$\mathbf{K}^* = \mathbf{K} - \mathbf{K}_i \tag{2.5}$$

where $\mathbf{K}^*$ is the stiffness matrix of the structure after the elements have been deleted and $\mathbf{K}_i$ is the stiffness matrix of the i^{th} element. Assuming that the removal of elements does not affect the load **f**, (2.6) shows how the displacement vector can be calculated.

$$\Delta\mathbf{u} = -\mathbf{K}^{-1}\Delta\mathbf{K}\mathbf{u} \tag{2.6}$$

Applying algebraic manipulation on (2.3) and (2.6):

$$\begin{aligned}\Delta\mathbf{C} &= \frac{1}{2}\mathbf{f}^{\mathrm{T}}\Delta\mathbf{u} \\ &= -\frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{K}^{-1}\Delta\mathbf{K}\mathbf{u} \\ &= \frac{1}{2}\mathbf{u}_i^{\mathrm{T}}\mathbf{K}_i\mathbf{u}_i\end{aligned} \tag{2.7}$$

where $\mathbf{u}_i$ is the displacement vector of the i^{th} element.

The sensitivity number (SN) of the mean compliance, α_i^e, can then be defined by (2.8).

$$SN = \alpha_i^e = \frac{1}{2}\mathbf{u}_i^{\mathrm{T}}\mathbf{K}_i\mathbf{u}_i \tag{2.8}$$

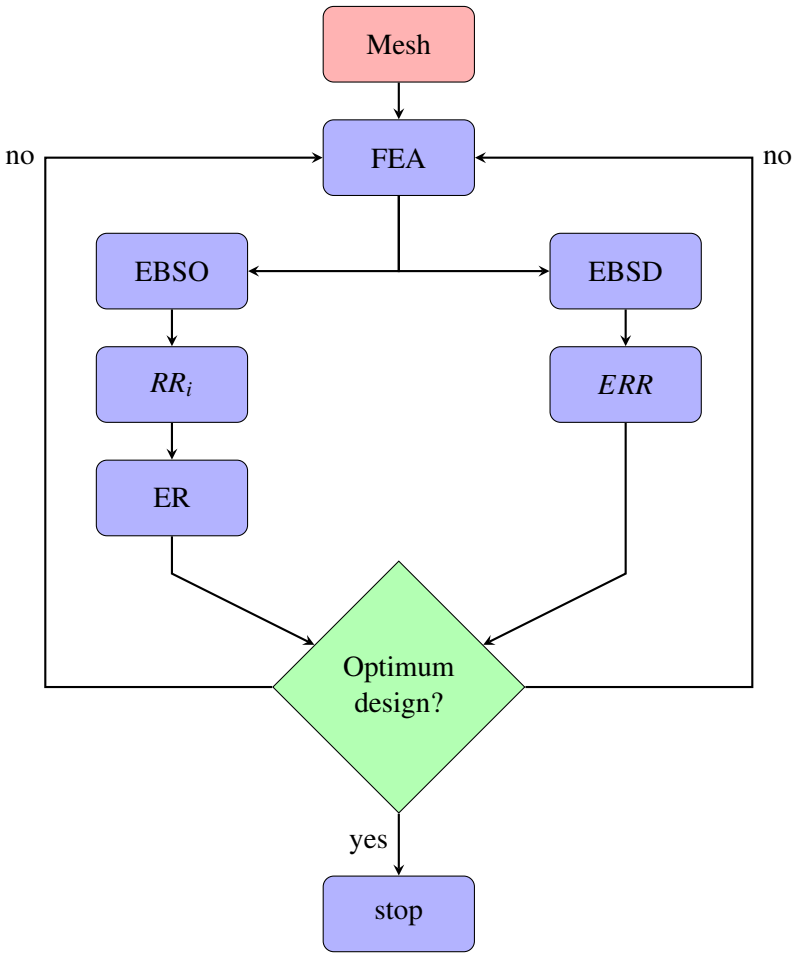


Figure 2.1: ESO method described in a flowchart

The equation (2.8) indicates that the increase of the mean compliance as a result of the removal of the i^{th} element is equal to its elemental strain energy. It is evident that the most effective and efficient way to remove elements to minimize the mean compliance, the equivalent of maximizing the stiffness, is to eliminate the elements which have the lowest values of α_i^e resulting in a minimal increase in $\mathbf{C}$. The number of elements to be removed at each iteration is determined by the element removal ratio (ERR),which is defined by the number of elements that can be deleted on each iteration, α_{del_max}, multiplied by the total numbers of elements on the benchmark mesh. It is important to say that there is no steady state specified in optimization for stiffness unlike the one based on stress level.

Figure 2.1 describes the EBSO and EBSD procedure.

Chapter 3

Bi-directional Evolutionary Structural Topology Optimization - BESO

Thinking about all types of topology optimization methods that were released up to now, the first and primordial difference between the Bi-directional Evolutionary Structural Topology Optimization (BESO) and them is the ability to remove and add material simultaneously. The primary research in 1999 [Yang et al., 1999] was focused on stiffness optimization.

From the displacement field calculated using FEA, the sensitivity number, SN, of all elements of the model is estimated using a linear extrapolation. The solid elements that have lower sensitivity numbers are removed from the domain, and the void elements that have higher sensitivity number are changed into solid elements. Every iteration two independent parameters determine the total number of elements removed and added. The first is the rejection ratio *RR* and the second one is the inclusion ratio *IR*.

The BESO method can be applied using stress criterion, von Mises [Querin et al., 2000]. In this case, solid elements with low stress are removed from the domain while void elements with high stress are switched into solid. The number of elements removed and added is defined by the rejection ration and inclusion ratio. The use of a parameter for rejection and another one for the addition of elements increases the cost of analysis. The selection of the values for *RR* and *IR* need to be done carefully or a good design will never be reached [Rozvany and G.I.N., 2009]. The first BESO algorithms also had another problem as:

- Low computational efficiency caused by the number of iterations
- Sometimes the final design is selected from a group of generated topologies
- The convergence history of the objective function is highly chaotic
- In continuum setting there is lack of solutions
- Mesh-dependency, with the use of a mesh with different sizes of elements, when new holes are introduced without adapt the structural volume will increase the efficiency of a specific design [Sigmund and Petersson, 1998, Bendsøe and Sigmund, 2003].

3.1 Problem Statement

In the early versions of the ESO/BESO methods, the objective and constraints were questionable, especially for the stiffness optimization cases. Considering that the initial goal was to search for the minimum material volume subject to determined mean compliance of displacement. The choice of inappropriate constraints or large element removal ratio imply in a worse solution. For the most common continuum structures the compliance-volume product tends to zero or unity. So, one way to fix this problem is the employment of a performance index using the compliance-volume product. In this case, the iteration process will stop when the index value drops fast. Furthermore, another problem may occur, the process could stop before a defined displacement constraint is satisfied.

The problem statement, volume constraint, is extensively used in topology optimization and is shown below:

$$\text{Objective function} \rightarrow \text{Minimize} \rightarrow \mathbf{C} = \frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{u} \tag{3.1}$$

$$\text{Constraints} \rightarrow V^* - \sum_{i=1}^{N} V_i x_i \tag{3.2}$$

$$\text{Design variable} \rightarrow x_i = \begin{cases} x_i = 0 \rightarrow void \\ x_i = 1 \rightarrow solid \end{cases} \tag{3.3}$$

where $\mathbf{f}$ and $\mathbf{u}$ are the load and displacement vectors and $\mathbf{C}$ is the mean compliance. V_i is the volume of the i^{th} element and V^* is the volume of the structure. N is the number of the elements in the domain. And the last one, x_i declares if an element is solid (1) or void (0).

The original ESO/BESO methods struggle to deal with the volume constraint statement. Since the volume is kept constant to satisfy the equation (3.1) to (3.3) the objective function may not converge. The primary goal of the new BESO method is to be stable convergent with the above problems statement.

3.2 Sensitivity Number - SN

Every time a solid element is deleted from the domain, the variation on the total strain energy is equal to the strain energy of the deleted element [Chu et al., 1996]. This variation is defined as the elemental sensitivity number, ESN:

$$\alpha_i^e = \frac{1}{2}\mathbf{u}_i^{\mathrm{T}}\mathbf{K}_i\mathbf{u}_i = \Delta\mathbf{C}_i \tag{3.4}$$

where $\mathbf{u}_i$ is the nodal displacement vector of the i^{th} element and $\mathbf{K}_i$ is the elemental stiffness matrix. The meshing process for complex models does not result in a uniform mesh. It means that the volume varies along the elements. In this case, the sensitivity number as presented by (3.4)

cannot be used. It does not consider the variation of the volume of each element of the nonregular mesh. To fix the obstacle, the *ESN* can be updated by dividing it by the volume, $\mathbf{V}_i$, of the i^{th} element, becoming elemental strain energy density *ESED* as:

$$\alpha_i^e = e_i = \frac{\left(\frac{1}{2}\mathbf{u}_i^{\mathrm{T}}\mathbf{K}_i\mathbf{u}_i\right)}{V_i} \tag{3.5}$$

From now on, ESED will be refereed as sensitivity number (SN) to simplify. After the calculation of the SN, the elements are organized in order from the lowest up to the highest value, $\alpha_{element}$ vector. Elements with low SN that are solid (1), will be transformed into voids (0), (3.6), and the void elements (0) that have high SN, it will be transformed into solids (1), (3.7).

$$\alpha_i \leq \alpha_{delete}^{th} \tag{3.6}$$

$$\alpha_i > \alpha_{adition}^{th} \tag{3.7}$$

The higher threshold is defined by $\alpha_{adition}^{th}$ while the lower is defined by α_{delete}^{th}. These limits will determinate if an element will be deleted or added in the design domain. The following rules determine how to calculate the thresholds.

1. $\alpha_{adition}^{th} = \alpha_{delete}^{th} = \alpha_{th}$

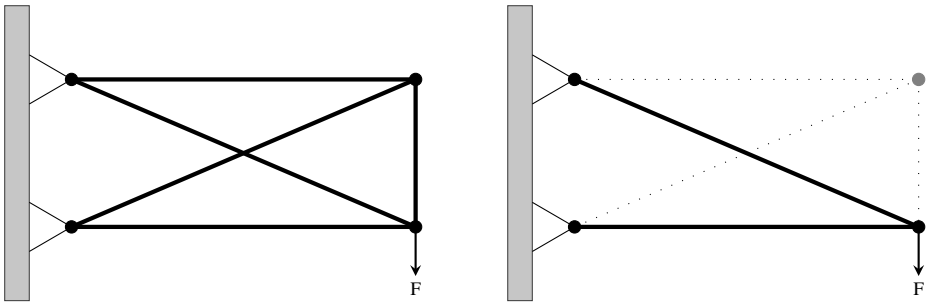
2. The target volume, V_{k+1}, is an upper boundary of the total number of solid elements in the design domains. For example, if the domains has 100 elements and the target volume corresponds to a domain with 75 elements, $\alpha_{th} = 75$

3. The volume addition ratio (AR), is the number of added elements divided by the total elements in the domain. AR_{max} is the maximum volume addition ratio. If $AR \leq AR_max$ otherwise recalculate $\alpha_{adition}^{th}$ and α_{delete}^{th}

4. Recalculating the thresholds. After organizing the sensitivity numbers of the void elements (0) the $\alpha_{adition}^{th}$ is calculated.

The number of elements will be equal to AR_{max} multiplied by the total number of elements in the design. α_{delete}^{th} is determined to respect to $V_{delete} = V_k - V_{k+1} + V_{added}$. AR_{max} controls the number of elements added in each iteration. Without this control, the structure may lose integrity when the BESO method starts from an initial guess design. Generally AR_{max} has a value higher than 1% to maintain the advantages of adding elements.

Using this plan of action, α_{add} controls the number of elements added in each iteration. Without this control, the structure may lose integrity when the BESO method starts form an initial guess design. Generally α_{add} has value higher than 1% to maintain the advantages of adding elements.



(a) Initial design (b) Optimized design

Figure 3.1: Example of an initial design before and after been optimized

3.3 Element Removal/Addition

The removal and addition of elements change the volume of the design along the optimization process. However, before it occurs, the target volume of the next iteration V_{k+1} needs to be calculated.

$$V_{K+1} = V_k\,(1 \pm EVR) \quad \text{where} \quad (K = 1,2,3,...) \tag{3.8}$$

Where EVR is the evolutionary volume ratio and k is the number of the current iteration. The volume constraint may be bigger or smaller than the initial design volume. So, the target volume may vary step by step. Once the volume goal is achieved, it will be maintained constant up to the end of the process.

$$V_{K+1} = V^* \tag{3.9}$$

3.4 Filter Scheme

The final product passed through the optimization process needs to be interpreted and manufacturable. Some inherit problem with the method is the checkerboard pattern problem. Usually, it occurs when low order finite elements are applied in continuum structures. It may happen in 2D and 3D elements. The sensitivity numbers could become discontinuous across elements. The Figure 3.2, presents a classic checkerboard pattern from the original ESO method.

The solution to this problem is the application of a smoothing scheme of averaging the sensitivity numbers of adjacent elements [Li et al., 2001], elemental filter (EF), Equation (3.10).

$$\alpha_i^e = \frac{\sum_{i=1}^{K} w(r_{ij})\alpha_j^e}{\sum_{i=1}^{K} w(r_{ij})} \tag{3.10}$$

$$w(r_{ij}) = r_{min} - r_{ij} \quad \text{where} \quad (j = 1,2...,k) \tag{3.11}$$

where K is the total number of elements in the domain, α_i^e the SN of the j^{th} element and r_{ij} is the distance between the center of the j^{th} element and the element analyzed.

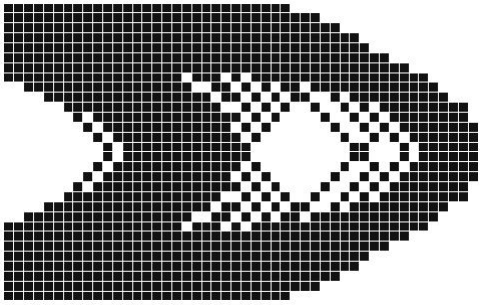


Figure 3.2: Checkerboard pattern

The nodal sensitivity number, need to be converted into smoothed elemental sensitivity numbers. The conversion is made by projecting the nodal sensitivity numbers to the design domain, Equation (3.12). The nodal filter (NF) uses a length scale r_{min} that does not vary with the mesh size variation.

$$\alpha_j^n = \sum_{i=1}^{M} w_i \alpha_i^e \tag{3.12}$$

$$w_i = \frac{1}{M-1}\left(1 - \frac{r_{ij}}{\sum_{i=1}^{M} r_{ij}}\right) \tag{3.13}$$

where M is the total number of elements connected to the j^{th} node, w_i is the weight factor of the i_{th} and $\sum_{i=1}^{M} w_i = 1$. r_{ij} is the distance between the center of the i^{th} element and the j^{th} node. The nodal sensitivity numbers does not have any physical meaning. After finish the projection, the filter uses the distance between the center of the j^{th} node and the element analyzed, Equation (3.15), to weight the influence of the nodal point concerning the analyzed element.

$$\alpha_i^n = \frac{\sum_{j=1}^{K} w(r_{ij})\alpha_j^n}{\sum_{j=1}^{K} w(r_{ij})} \tag{3.14}$$

$$w(r_{ij}) = r_{min} - r_{ij} \quad \text{where} \quad (j = 1, 2..., k) \tag{3.15}$$

Here, the main idea is to use the elements that are inside the area of the filter to calculate improved sensitivity number (ISN) of the i^{th} element. The Figure 3.3 shows the graphic representation of the filter. The circle of radius r_{min} is centered in the centroid of the i^{th} element. The size of the domain needs to be bigger enough to cover more than one element.

The nodal filter scheme, (3.14), does not take in account if the element is solid or void. At the beginning of the process, the void elements have *ISN* set to zero by default. Meanwhile, with the iterations evolving the void elements starts to received nonzero *ISN* numbers. Therefore, the algorithm will be able to calculate the importance of both types of elements ranking them along the structure.

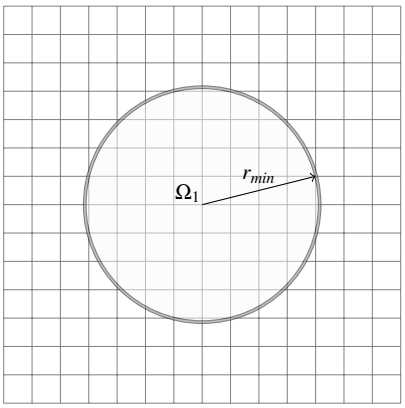


Figure 3.3: Filter scheme

Another drawback that happens with the original ESO is the mesh-dependency problem. It generates different topologies when different meshes are used. Using a refined mesh, with smaller elements, the method will produce a resultant topology with members with smaller sizes. Changing the size of the mesh elements will result in different topologies qualitatively or more detailed ones. Ideally, after refinement of the mesh, the process should result in a better finite element modeling of the same optimal structure and with a better description of the boundaries.

Some solutions can be applied to solve the mesh-dependency problem like the perimeter control method [Yang et al., 2003, Haber et al., 1996, Jog, 2002] or the sensitivity filter scheme [Sigmund, 1997, Sigmund and Petersson, 1998]. Perimeter control applied to BESO method can find the mesh-independent solution. The reason is the use of the perimeter length on it. Although defining the best value of the perimeter length can be hard.

3.5 BESO Procedure

Figure 3.4 presents a schematic overview of the BESO method. It begins with de initial definitions as design domain, loads, boundary conditions, restrictions, and FE mesh. In the second step the BESO parameters, V^*, ER and AR_{max}, are selected. The third step is where the numeric simulation happened. The following step is the calculation of the sensitivity number for each element. Then the filter scheme is applied, and sequentially the average of the sensitivity number is calculated. After this, the target volume is calculated for the next iteration, and the new design is constructed. At this part of the process, the algorithm will verify if the volume constraint is satisfied. If it is not satisfied, the next iterations returning to the FEA step starts. Otherwise, it will verify if the model converged. If it has not converged, it will return to the third step. In case both verifications are positive, the algorithm reaches the objectives and stops the process.

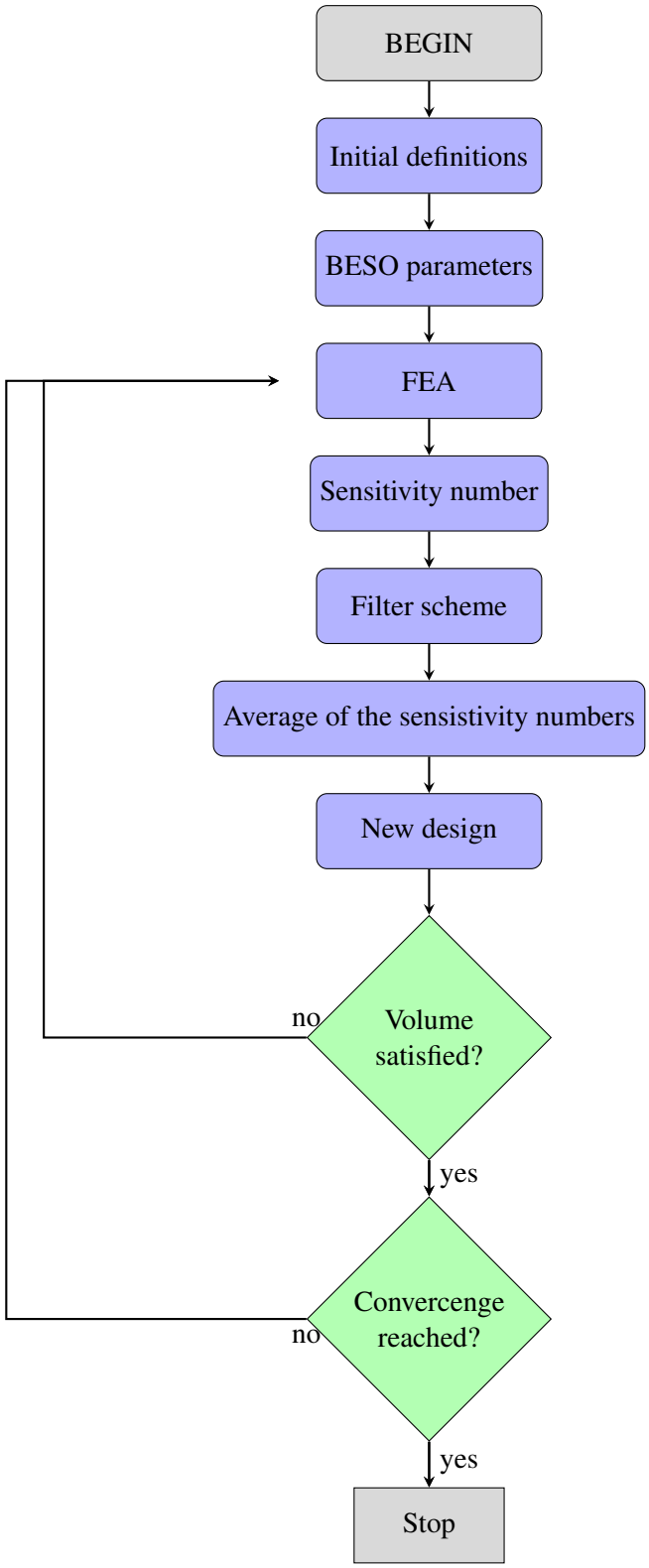


Figure 3.4: BESO Method

Chapter 4

Continuum Mechanics

The fundamentals of Solid Mechanics has been largely studied for the last decades. The number of publications and articles related to this topic is enormous. This chapter is based on the following well-known reference: Timoshenko and Goodier [1971]. For a given solid and boundary conditions, forces and constraints, the relation between stress and strain, and strain and deformation is the base of the solid mechanics.

Some initial definitions and assumptions:

Stress: express the loading in terms of force applied to a cross-sectional area of an object, given by (4.1).

$$\sigma = \frac{F}{A_0} \tag{4.1}$$

Stress Field: is the distribution of internal forces that balance a set of external loads. Occurs when a set of loads is applied to a structure.

Strain: is the response of a system to applied stress, given by (4.2). The stress causes the material to deform in a linear behavior, so the stresses are linearly related to the strains.

$$\varepsilon = \frac{\Delta L}{L_0} \tag{4.2}$$

Elastic behavior: when the application of loads deforms a solid, it returns to the initial, undeformed, shape when the loads cease.

4.1 The state of stress at a point

Since the stress is the principal variable of solid mechanics, the state of stress at a point is given by the stress tensor, (4.3). It also can be written in a vector form, Voigt notation, (4.4), where σ is

the normal stress and τ is the shear stress.

$$\sigma = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{bmatrix} \tag{4.3}$$

$$\sigma = \begin{Bmatrix} \sigma_{xx} & \sigma_{yy} & \sigma_{zz} & \tau_{xy} & \tau_{yz} & \tau_{zx} \end{Bmatrix}^{\mathrm{T}} \tag{4.4}$$

4.2 Equilibrium

If a solid is subjected to a set of loads, the stress distribution needs to be compatible with the global equilibrium of the body. Also, the equilibrium must be ensured in every part of the system. Considering an infinitesimal element from this body and solving its equations, the result is the equilibrium equations of elasticity, (4.5).

$$\begin{aligned} \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{xz}}{\partial z} + F_x = 0 \\ \frac{\partial \tau_{yx}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + F_y = 0 \\ \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} + F_z = 0 \end{aligned} \tag{4.5}$$

Using σ as the stress tensor, ∇ as the gradient and $\mathbf{F}$ as the body force vector, the system of equations can be reduced to (4.6).

$$\nabla \sigma + \mathbf{F} = 0 \tag{4.6}$$

4.3 Strain and displacement

In the small deformation theory, a normal strain, ε, is the change in the length of a line segment between two points divided by the original length of the line segment. Meanwhile, a shearing strain, γ, is the angular change between two line segments which were initially perpendicular. The relation between strains and displacements is ruled by (4.7).

$$\begin{aligned} \varepsilon_{xx} = \frac{\partial u}{\partial x}, \quad \gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \varepsilon_{yy} = \frac{\partial v}{\partial y}, \quad \gamma_{yz} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \\ \varepsilon_{zz} = \frac{\partial w}{\partial z}, \quad \gamma_{zx} = \frac{\partial w}{\partial x} + \frac{\partial u}{\partial z} \end{aligned} \tag{4.7}$$

where u, v and w are the respective displacements in the x, y and z directions. Remenbering that:

$$\begin{aligned} \gamma_{xy} &= \gamma_{yx} \\ \gamma_{yz} &= \gamma_{zy} \\ \gamma_{xz} &= \gamma_{zx} \end{aligned} \tag{4.8}$$

The equations (4.7) are known in the solid mechanic as the cartesian strain displacement relationships. They can be written in the matrix form as follows:

$$\varepsilon = \mathbf{L}\mathbf{u} \tag{4.9}$$

where $\mathbf{L}$ and $\mathbf{u}$ are:

$$\mathbf{L} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial z} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}^{\mathrm{T}} \tag{4.10}$$

$$\mathbf{u} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{4.11}$$

Since a tensor describes the state of stress at a point, the same happens with the strain, as (4.12) and in vector form (4.13)

$$\varepsilon = \begin{bmatrix} \varepsilon_{xx} & \gamma_{xy} & \gamma_{xz} \\ \gamma_{yx} & \varepsilon_{yy} & \gamma_{yz} \\ \gamma_{zx} & \gamma_{zy} & \varepsilon_{zz} \end{bmatrix} \tag{4.12}$$

$$\varepsilon = \begin{Bmatrix} \varepsilon_{xx} & \varepsilon_{yy} & \varepsilon_{zz} & \gamma_{xy} & \gamma_{yz} & \gamma_{zx} \end{Bmatrix}^{\mathrm{T}} \tag{4.13}$$

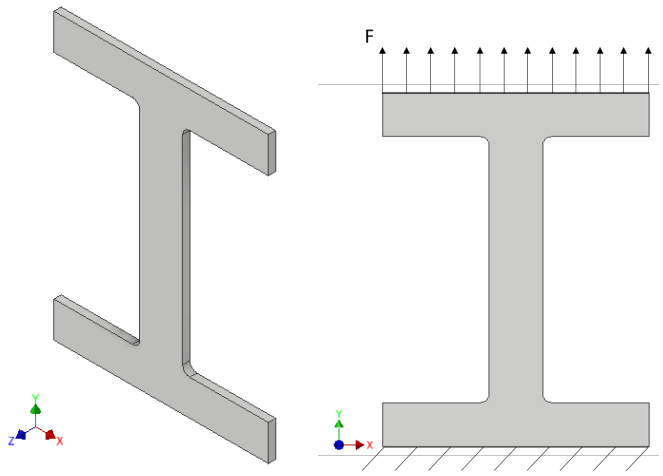
4.4 Constitutive relations

The material applied in all cases studied in this thesis was considered isotropic and homogeneous. The elastic properties are defined by Young's modulus E, and Poisson's ratio ν. Hooke's laws define the existent relationship between stress and strain, (4.14).

$$\sigma = \mathbf{C}\varepsilon \tag{4.14}$$

where $\mathbf{C}$ is the stiffness matrix, it is an asymmetric non-singular matrix, resulting in:

$$\varepsilon = \mathbf{S}\sigma \tag{4.15}$$



(a) (b)

Figure 4.1: Plane stress

where $c = \mathbf{S}^{-1}$, is given by

$$s = \frac{1}{E}\begin{bmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(1+\nu) & 0 & 0 \\ 0 & 0 & & 0 & 2(1+\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(1+\nu) \end{bmatrix} \tag{4.16}$$

4.5 Plane stress problems

When analyzing elasticity problems in which a body is subjected to a force applied in a specific plane, and its body has the thickness smaller than the other dimensions, we can say it is a plane stress problem, as presented in the Figure 4.1. In this particular case, we have:

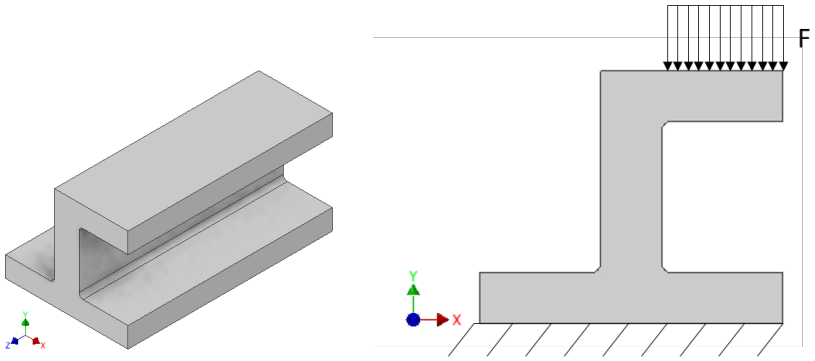
$$\sigma_{zz} = \tau_{xz} = \tau_{yz} = 0 \tag{4.17}$$

resulting in new stress and strain vectors:

$$\sigma = \begin{Bmatrix} \sigma_{xx} & \sigma_{yy} & \tau_{xy} \end{Bmatrix}^{\mathrm{T}} \quad \text{and} \quad \varepsilon \begin{Bmatrix} = \varepsilon_{xx} & \varepsilon_{yy} & \gamma_{xy} \end{Bmatrix}^{\mathrm{T}} \tag{4.18}$$

The stiffness matrix $\mathbf{C}$ becomes:

$$c = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \tag{4.19}$$



(a) (b)

Figure 4.2: Plane strain

4.6 Plane Strain problems

When the thickness is no longer smaller than the others dimensions or the body has one of its dimensions larger than the others and the cross section does not vary along this direction, we have a plane strain problem, as presented in Figure 4.2. Considering that the applied loads are in the *xy*-plane, the equations can be reduced to:

$$w = 0 \quad \text{and} \quad \frac{\partial}{\partial z} = 0 \tag{4.20}$$

.
resulting in:

$$\varepsilon_{xx} = \varepsilon_{yy} = \gamma_{xy} = 0 \tag{4.21}$$

Then, the material matrix c is reduced to :

$$c = \begin{bmatrix} \lambda + 2G & \lambda & 0 \\ \lambda & \lambda + 2G & 0 \\ 0 & 0 & G \end{bmatrix} \tag{4.22}$$

Here, λ is the Lamé coefficient, and G is the modulus of rigidity. These constants are related with E and ν by:

$$G = \frac{E}{2(1+\nu)} \quad \text{and} \quad \lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} \tag{4.23}$$

Chapter 5

Finite Element Method - FEM

5.1 FEM formulation

One of the keys factors to apply any optimization process is the stiffness. Remembering that the most common goals are to reduce the weight and or increase the stiffness. Moreover, generally to achieve the desired stiffness the mean compliance is used, (5.2). As this works is focused on optimization, FEM formulation will not be presented, if the reader desire more information about it, the author recommends some popular material for research like: Fish and Belytschko [2007] and some related articles, Clough and Wilson [2000], Johns [1953].

$$F = K\mathbf{u} \tag{5.1}$$

$$C = \frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{u} \tag{5.2}$$

5.2 Classic Element - C3D8

FEM has an extensive element library to provide a robust set of tools for solving many different problems. In order to implement the optimization algorithm in SOLIDFEM, the classic eight-node brick element, C3D8, was chosen in this work. Each element has aspects that characterize its behavior: family, degrees of freedom, number of nodes, formulation, and integration. This chapter was based on the following well-known reference: [Simulia, 2014, Mirmiran et al., 2007, DS SIMULIA Corp, 2011]

A family of finite elements is the broadest category used to classify elements. Elements in the same family share many basic features, and there are many variations within a family. Figure 5.1 show some examples of family types.

The degrees of freedom is the fundamental variables calculated during the analysis. For a stress/displacement simulation the degrees of freedom are the translations and, for shell, pipe and beam elements, the rotations at each node. For a heat transfer simulation the degrees of freedom are the temperature at each node; for a coupled thermal-stress analysis temperature degrees of

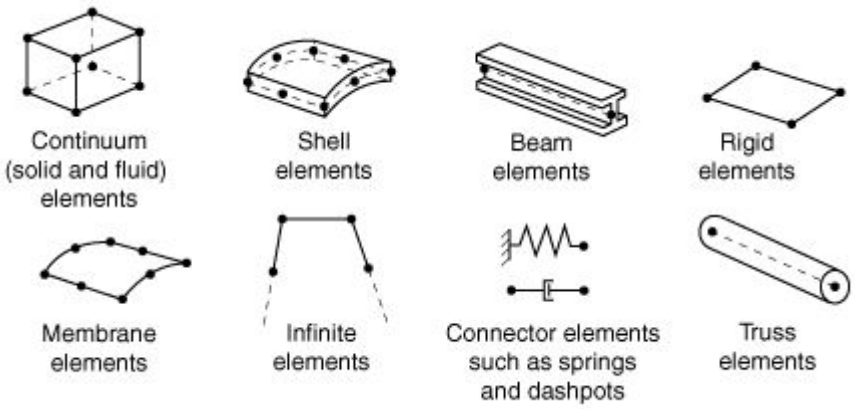


Figure 5.1: Element family, from Simulia [2014]

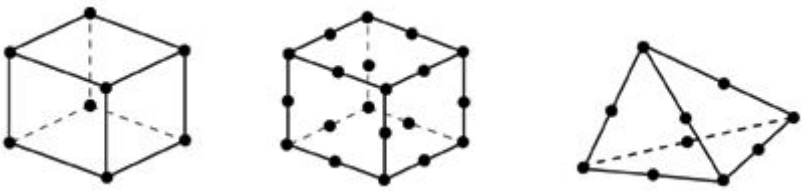
freedom exist in addition to displacement degrees of freedom at each node. Heat transfer analyses and coupled thermal-stress analyses, therefore, require the use of different element than does a stress analysis since the degrees of freedom are not the same.

Displacement or other degrees of freedom are calculated at the nodes of the element. At any other point in the element, the displacements are obtained by interpolating from the nodal displacements. Usually, the interpolation order is determined by the number of nodes used in the element.

- Elements that have nodes only at their corners, such as the 8-node brick shown in Figure 5.2 (a), use linear interpolation in each direction and are often called linear elements or first-order elements.
- Elements with mid-side nodes, such as the 20-node brick shown in Figure 5.2(b), use quadratic interpolation and are often called quadratic elements or second-order elements
- Modified triangular or tetrahedral elements with mid-side nodes, such as the 10-node tetrahedron shown in Figure 5.2(c), use a modified second-order interpolation and are often called modified or modified second-order elements.

An element's formulation refers to the mathematical theory used to define the element's behavior. In the Lagrangian or material, the elements deform with the material. In the alternative Eulerian, or spatial, the elements are fixed in space as the material flows through them. Eulerian methods are used commonly in fluid mechanics simulations and, stress/ displacement uses Lagrangian formulation.

There are various numerical techniques to integrate quantities over the volume of each element, thus allowing complete generality in material behavior. Using Gaussian quadrature is possible evaluating the material response at each integration point in each element. Some continuum elements can use full or reduced integration, a choice that can have a significant effect on the accuracy of the element for a given problem. In this work, the reduced integration was not used.



(a) Linear element (8-node brick, C3D8)

(b) Quadratic element (20-node brick, C3D20)

(c) Modified second-order element (10-node tetrahedron, C3D10M)

Figure 5.2: Linear brick, quadratic brick, and modified tetrahedral elements, from Simulia [2014]

Chapter 6

Algorithms

Along with this work the ESO and BESO methods were presented. All the formulation and characteristics of these methods were based on the literature, Chapter 2, and 3. However, the process of implementation of the code resulted in modified and simplified versions of the fundamental algorithms. It may have happened because of the complexity of the Fortran language, programming difficulties and necessity to start from a simple version of each algorithm to truly understand it. The most significant distinction from the literature and our version is initially the way the algorithms calculate the thresholds, no use of additional filters to ensure the stability and or convergence.

The higher threshold is defined by $\alpha^{th}_{adition}$ while the lower one is defined by α^{th}_{delete}. These limits determinate if an element will be deleted or added in the design domain. The number of elements deleted on each iteration is defined by α_{del} and the elements added by α_{add}. The following rules determine how the thresholds are found

1. On the first iteration, the threshold for the elements deletion is the *SN* of the element on the position α_{del} of the vector $\alpha_{element}$:

$$\alpha^{th}_{delete} = \alpha_{element}(\alpha_{del}) \tag{6.1}$$

2. Also on the first iteration, the threshold for element addition is the *SN* of the element on the position α_{add} of the vector $\alpha_{element}$:

$$\alpha^{th}_{addition} = \alpha_{element}(\alpha_{add}) \tag{6.2}$$

3. For the next iteration (IT) the positions will be renewed by the expressions:

$$\alpha_{element}(\alpha_{del} * IT) \quad \text{and} \quad \alpha_{element}(\alpha_{add}/IT) \tag{6.3}$$

The Figure 6.1 shows the iteration 1 and 2 of ESO and BESO process. ESO and BESO deletion are represented by the red block, and the blue block represents BESO addiction. he main difference between both algorithms is the number of deleted elements from the mesh on each iteration. All

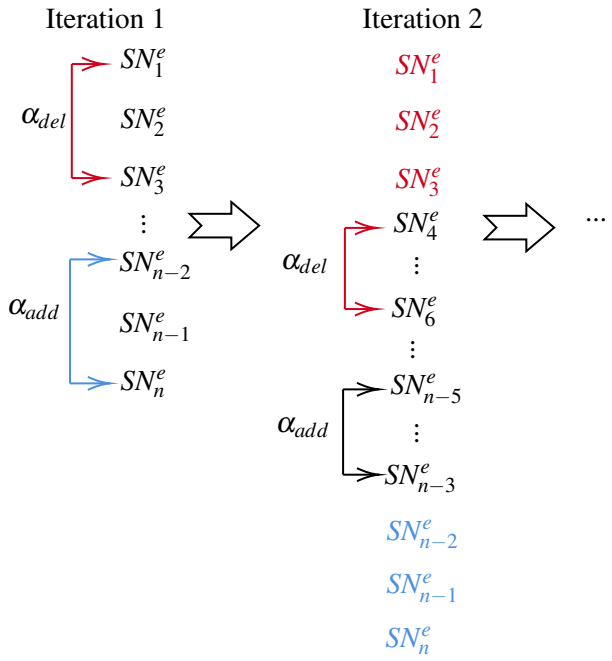


Figure 6.1: Sensitivity number flowchart for ESO and BESO methods

methods use the variable α_{del} as a control parameter, limiting the number of elements transformed in voids on every iteration.

The simulation using the ESO 1 method follows a linear equation, (6.4) to vary the value of α_{del} along the process. The ESO 2 method uses a constant value for all iterations, and the ESO 3 also initialize the process using a constant value for α_{del}, but it has an evolutionary parameter, α_{adj}, that update the variable value when the optimization process (OP) became stable. Here the *OP* is considered steady when it does not delete any element for some previously defined iterations. In (6.5) the logical process used by the ESO 3 method is presented. The BESO method uses the same structure of the ESO 2, but it is slower because of the addition of elements along the simulations.

$$N_{deleted} = \alpha_{DEL} * IT \tag{6.4}$$

$$\text{Process stable?} \rightarrow \begin{cases} \text{yes} \rightarrow \alpha_{ADJ} = \text{new value} \rightarrow N_{deleted} = \alpha_{DEL} + \alpha_{ADJ} \\ \text{no} \rightarrow \text{continue} \end{cases} \tag{6.5}$$

As the algorithms delete elements in distinct ways and amounts on each iteration, as a consequence, the speed to achieve the optimum design likewise differs. Figure 6.2 displays all variables α_{del} in a fictional case with 100 iteration. The ESO 1 method is the fastest one, followed by the ESO 2, although it was expected that the ESO 3 method would be the second one, it needs more iterations to achieve the same reduction, since it became stable along the process, and depending on the criterion used to update α_{adj}, the computational cost is increased. The BESO method is based on ESO 2, but the BESO is slower than the ESO 2 because there is an addition of elements

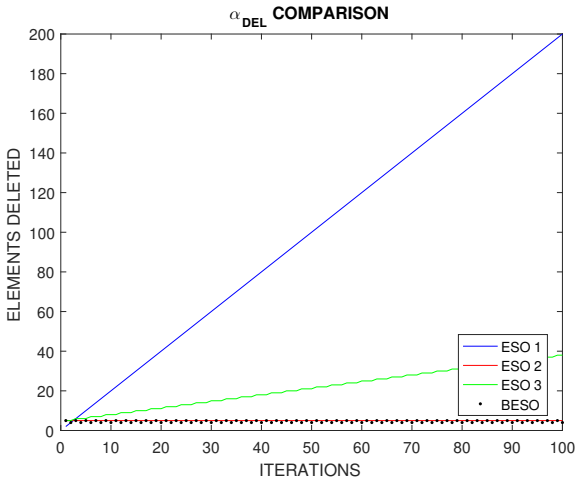


Figure 6.2: Differences between algorithms when deleting elements

when the related standards are achieved.

The criteria used to add elements on the BESO method is based on the idea that when one element is deleted, it creates spikes of *SN* on the remaining solid elements around it. If the variation on *SN* is higher than a predefined value, this void element is reactivated to decrease the mean *SN* of that specific area on the mesh. The conditions applied to the addition criteria can be seen below

- Void Elements surrounded by more than two solid ones, sharing edges.
- Variation of *SN* on a void element higher than 3% between sequential iterations
- Variation of *SN* on the solid neighbors higher than 5% between sequential iterations

These conditions were defined in order to guarantee that on each iteration some predefined amount of elements should be re-activated. Here, the main idea is to force the reactivation of at least one element every iteration.

Chapter 7

Cases and Methodology

7.1 Cases

As the number of optimization methods and implementations of these methods increases, the use of comparative studies to evaluate their performance is useful. In order to realize these studies and further comparisons, some well-known benchmarks for optimization process were used. Figure 7.1 shows each case. The Table 7.1 presents the name of each benchmark and their abbreviation that were used along with this work. The cases used in this work can be found in Beiranvand et al. [2017], Rojas-Labanda and Stolpe [2015], Labanda and Stolpe [2014], Huang and Xie [2010a]. The nomenclature used for all cases followed the rule presented in Figure 7.2.

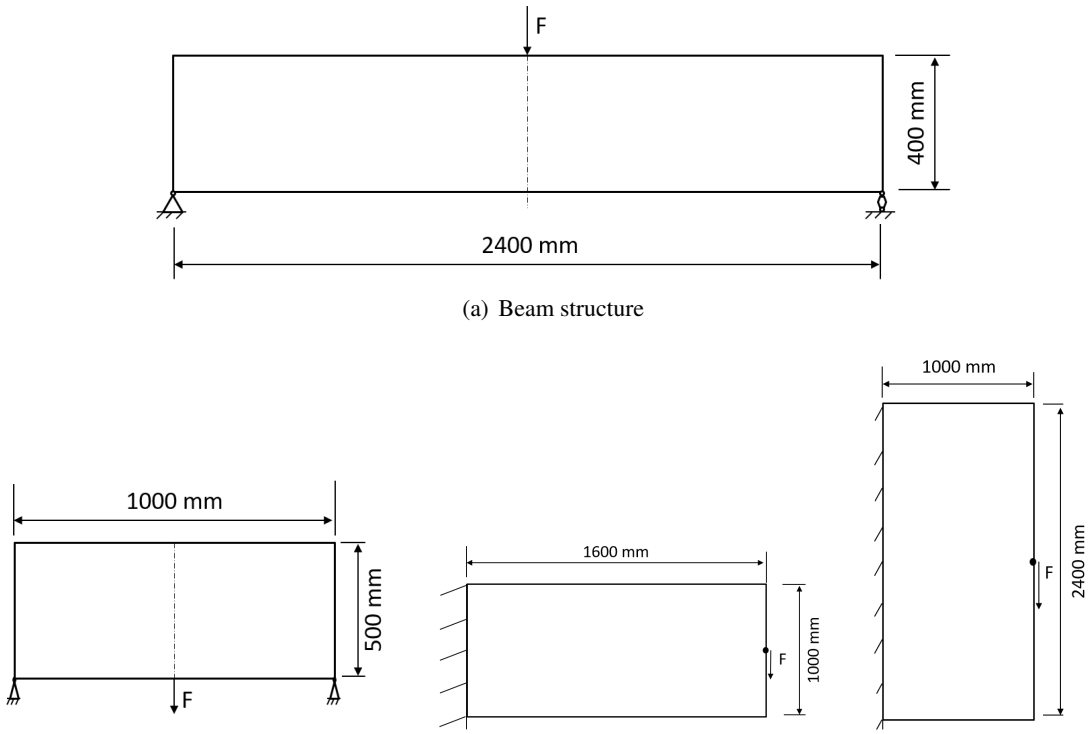
Table 7.1: Benchmark cases and their abbreviations

CASES	MESH	ABBREVIATION
Beam structure	120X20	BS_120X20
Michell type structure	50X25	MTS_50X25
Short cantilever	48X30	SC_48X30
Two bar frame	25X60	TBF_25X60

The characteristics measures as width W, height H, thickness t, Young's modulus E, Poisson ration ν and load F were set to International System of Units (SI) to uniform all the data and results. The Table 7.2 describes all the parameters used.

Table 7.2: Benchmarks parameters

CASES	W [mm]	H [mm]	t [mm]	E [MPa]	ν	F [N]
BS_120X20	2400	400	1	2.00E+05	0.3	20000
MTS_50X25	1000	500	1	1.00E+05	0.3	1000
SC_48X30	1600	100	1	2.07E+05	0.3	3000
TBF_25X60	1000	2400	1	1.00E+05	0.3	400



(a) Beam structure

(b) Michell type structure

(c) Short cantilever

(d) Two-bar frame

Figure 7.1: Cases used in the work. [Huang and Xie, 2010a]

7.2 Methodology

In order to properly evaluate and compare all methods, some standards were defined to uniform the analysis. A list of criteria is presented below. In addition to these criteria, the reduction imposed was fixed to 75% of the initial volume of de model, for all cases.

- Performance Index (*PI*)
- Volume versus mean compliance
- Total simulation/process time
- Number of iterations up to the optimum design

The *PI* graphic was built using (7.1), where σ_{VMe} is the element von Mises stress, averaged from the element Gauss points, V_e is the element volume, F is a representational force applied, from Table 7.2, and L is a reference length of the model, the biggest dimension of the model. Here, lower values of *PI* imply a better design.

$$PI = \frac{\sum_{elements} \sigma_{VMe} V_e}{FL} \tag{7.1}$$

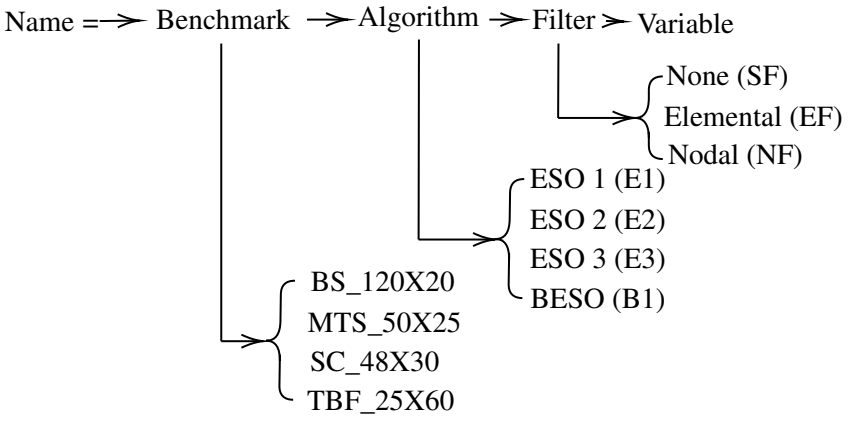


Figure 7.2: Nomenclature creation rule

The *volume versus mean compliance* graphic is used to find the relation between the reduction of the volume with the increase of the mean compliance while the elements are deleted or deleted and added.

All criterion shows the evolution of the design along the simulation since here the time step is the actual iteration of the process

7.3 Studies

The research realized on this thesis was divided into the following steps:

- Study of the influence of the α_{del_max} variable
- Influence of the additional variable applied to BESO algorithm
- Verification of the occurrence of checkerboard pattern and mesh dependency
- Application and comparison between elemental and nodal filters and influence of the filters radius
- 3D application
- Efficiency of the evolutionary methods

The first study performed was check how the variable α_{del_max}, that controls the maximum number of elements deleted in each iteration of the optimization process, influences the final design and performance of the case study. The primary goal is to acquire de knowledge to set the best value of this variable for each case and, if possible, one value for all of them. In other words, fulfill a sensitivity test.

All the α_{del_max} study cases were simulated following the Table 7.3, where each case is shown with its respective initial number of elements. For each α_{del_max} value there is one respective number of deleted elements for each case, notice that for some cases, the number of elements

deleted is the same, the numbers in bold. It happens because the algorithm needs to delete an even number each time or the result may not be symmetric. For this reason, the number of elements to be deleted was set to be always an even number. All the case were simulated using the ESO 1, ESO 2 and ESO 3 methods. All cases abbreviations for this study can be found in Appendix A.

Table 7.3: α_{del_max} study

	α_{del_max}	BS_120X20 2400	MTS_50X25 1250	SC_48X30 1440	TBF_25X60 1500
1	0.0010	2	2	2	2
2	0.0020	4	4	2	2
3	0.0030	8	4	4	4
4	0.0040	10	6	6	6
5	0.0050	12	8	8	6
6	0.0075	18	12	10	10
7	0.0100	24	16	14	12
8	0.0125	30	18	18	16
9	0.0150	36	22	22	18
10	0.0175	42	26	26	22
11	0.0200	48	30	28	26

For each case simulated, the final design was compared with the others α_{del_max} values along with the particular *PI* graphic. This should show the best α_{del_max} for each simulated case.

The second study is the influence of the additional variable applied to BESO algorithm. This study searches the best value of the BESO addition index (I-ADD-BESO), it controls the maximum number of elements added on each iteration. Knowing that the algorithm BESO was based on the ESO 2, its α_{del_max} best value was considered for BESO algorithm. The Table 7.4 presents how the study was performed. Also, the nodal and elemental filter was applied with a radius of $60\,mm$, since the method needs to be used with the filters active. The numbers in bold indicate that their values are the same as the previous case. All the cases abbreviations for this study can be found in Appendix B.

The third study, the verification of the occurrence of checkerboard pattern (CP) and mesh dependency (MD), was realized by running all optimization methods with each benchmark case. In this study, the results were based on the visual result for every simulation. The mesh used in each case for this topic are presented in the Table 7.5. The table with all the cases abbreviations for this study can be found in appendix C.

For the fourth study, the elemental and nodal filters were used with every method and applied to all benchmarks. Now, the goal é verify the efficiency of both filters comparing the total time, number of iterations and final design. The table with all the cases abbreviations for this study can be found in the annex B. Simultaneously, the fifth study, the influence of the radius on the filters results, were realized by varying the value of the filter radius from the minimum up to three times the value of the element size of for each benchmark. The volume reduction was set to 50% of the initial volume since the objective here was to evaluate the filter, reductions over this value would

Table 7.4: Addition study for BESO

I-ADD-BESO	BS_120X20 2400	MTS_50X25 1250	SC_48X30 1440	TBF_25X60 1500
0.001	2	2	2	2
0.002	4	2	**2**	**4**
0.003	8	4	4	4
0.004	10	6	6	6
0.005	12	6	8	8
0.006	14	8	**8**	10
0.007	16	8	10	**10**
0.008	20	10	12	12
0.009	22	12	**12**	14
0.010	24	12	14	**16**

only increase computational cost. Table 7.6 presents the radius variation for each benchmark and Appendix D presents all the cases abbreviations for this study.

The fifth study, a 3D application, has the purpose of proving that the evolutionary algorithms work properly with 3D models. The discretization used for each benchmark is shown below and correspond to:

- BS-120X20X10-E1-NF-25
- MTS-50X25X10-E2-EF-25
- SC-48X30X10-E3-NF-50
- TBF-25X60X10-B1-EF-95

The last study, the efficiency of the evolutionary methods, has the mission to compare all the algorithm used in this work and show which one has the best relation between computational cost, time and design efficiency for each case. Its structure is presented at Table **??**. All the cases abbreviations for this study can be found in Appendix F. The total number of simulations is presented in the Table 7.8.

Table 7.5: Checkboard and mesh dependency

BS_60X40	MTS_50X25	SC_32X20	TBF_25X60
BS_120X20	MTS_80X40	SC_48X30	TBF_50X120
BS_180X60	MTS_100X50	SC_64X40	TBF_75X80

Table 7.6: Radius for each benchmark

	RADIUS [mm]			
	BS_120X20	MTS_50X25	SC_48X30	TBF_25X60
1	20	20	35	40
2	25	25	40	45
3	30	30	45	50
4	35	35	50	55
5	40	40	55	60
6	45	45	60	65
7	50	50	65	70
8	55	55	70	75
9	60	60	75	80
10	65	65	80	85
11			85	90
12			90	95
13			95	100
14			100	105
15			105	110
16				115
17				120

Table 7.7: Comparison between algorithms

	ALGORITHM	FILTER
BENCHMARKS	ESO 1	NONE ELEMENTAL NODAL
	ESO 2	NONE ELEMENTAL NODAL
	ESO 3	NONE ELEMENTAL NODAL
	BESO	NONE ELEMENTAL NODAL

Table 7.8: Total number of simulations

TESTS	SIMULATIONS
α_{del_max}	132
BESO addition	40
CH and MD	36
Filters	504
3D	4
Final comparison	44
TOTAL	760

Chapter 8

Results and Discussions

8.1 Influence of the Eldelmax - α_{del_max}

All benchmarks were analyzed using PI and the performance index normalized (PI-NORM) plots, and the correspondent maximum alpha deletion variable (Eldemax), α_{del_max}, that results in the lowest *PI* is presented in the Figure 8.1 for the BS-120X20 case. The others results can be found in Appendix A. With the best values defined for the variable α_{del_max}, the next study was simulated using them.

The results for α_{del_max} study are presented in the Table 8.1. Initially, the results reveal differences when the benchmark is changed, as can be seen in the SC_48X30 case. In order to facilitate the next stages of this work, just one α_{del_max} value was chosen for each algorithm. The selected values are the bold ones in the α_{del_max} column in Table 8.1.

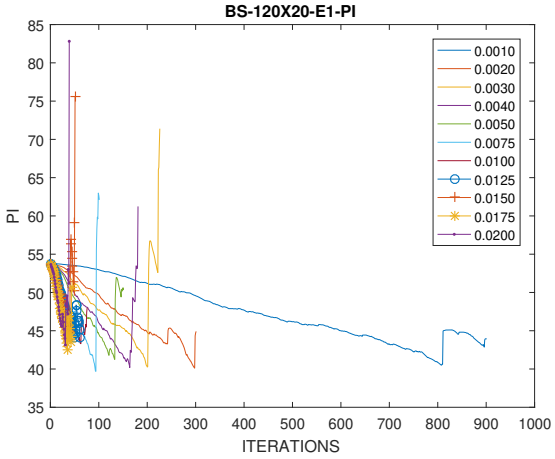
The decision process was done by analyzing the results for each case using the *PI* plot. For each algorithm, there is the particular *PI* plot,

8.2 Influence of the addition variable

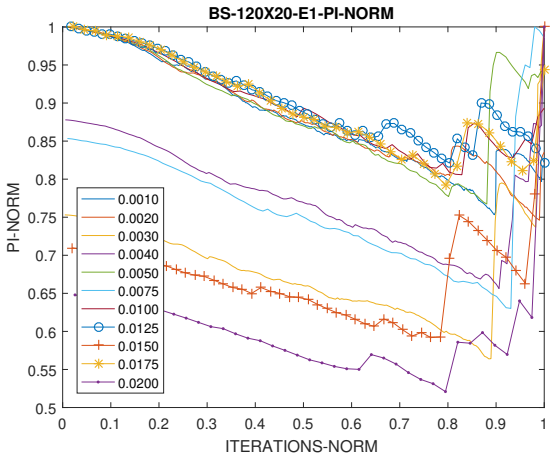
Since the BESO algorithm was based on ESO 2 code, the best value of α_{del_max} for ESO 2, 0.0200 from Table 8.2, was used to run the BESO simulations. In order to run this test, the filter radius was set to 60 mm for all cases. The results for each benchmark is presented in Table 8.2. The best value for each algorithm is shown at α_{del_max} column. Figure 8.2 presents the results for case MTS-50X25 using elemental and nodal filters. The others results can be found in Appendix B.

Table 8.1: α_{delmax} results for each benchmark

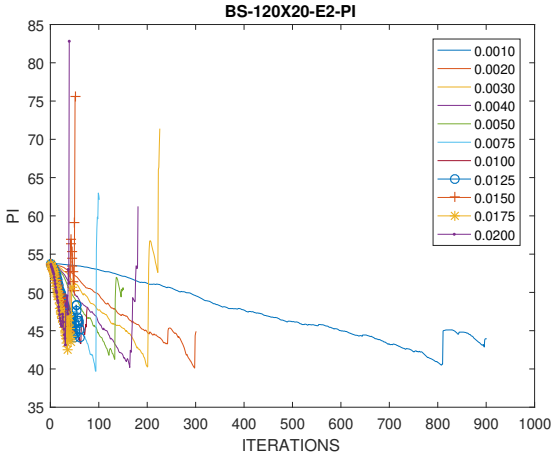
	BS_120X20	MT_50X25	SC_48X30	TBF_25X60	α_{delmax}
ESO 1	0.0200	0.0200	0.0010	0.0200	**0.0200**
ESO 2	0.0200	0.0200	0.0010	0.0200	**0.0200**
ESO 3	0.0030	0.0030	0.0500	0.0100	**0.0030**



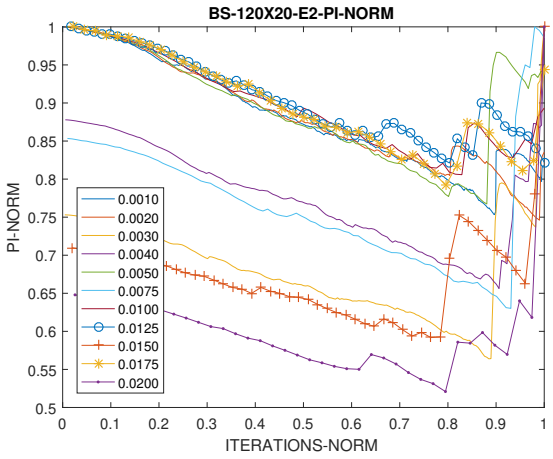
(a) BS-120X20-E1 PI



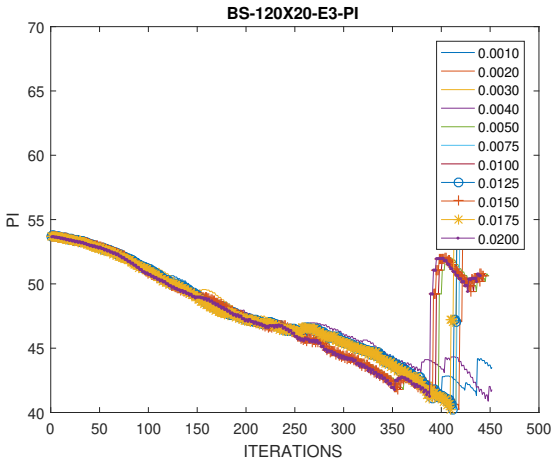
(b) BS-120X20-E1 PI normalized



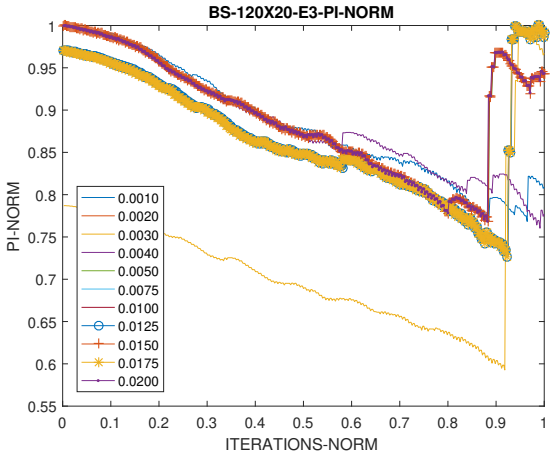
(c) BS-120X20-E2 PI



(d) BS-120X20-E2 PI normalized



(e) BS-120X20-E3 PI

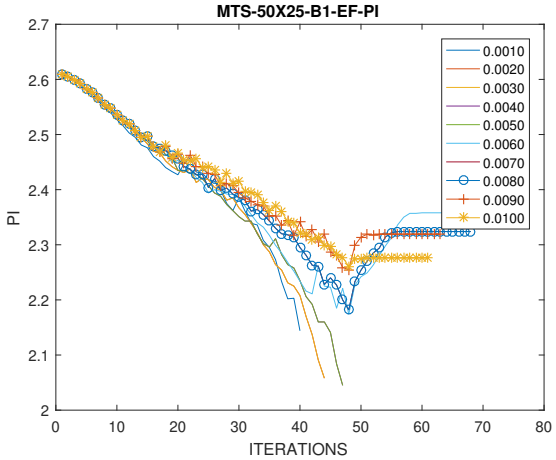


(f) BS-120X20-E3 PI normalized

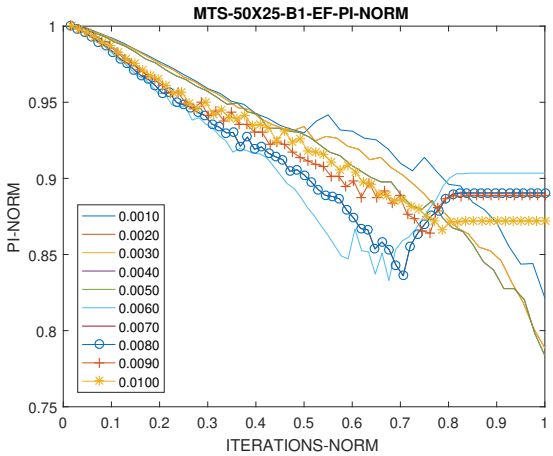
Figure 8.1: α_{del_max} results

Table 8.2: Influence of the addition variable

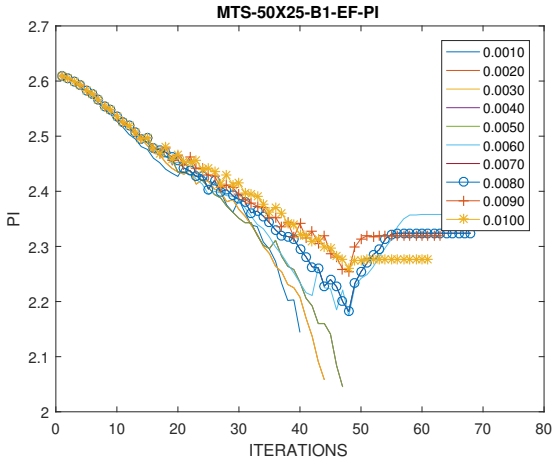
	BS-120X20	MT-50X25	SC-48X30	TBF-25X60	α_{del_max}
BESO EF	0.001	0.006	0.006	0.006	**0.006**
BESO NF	0.001	0.006	0.004	0.006	**0.006**



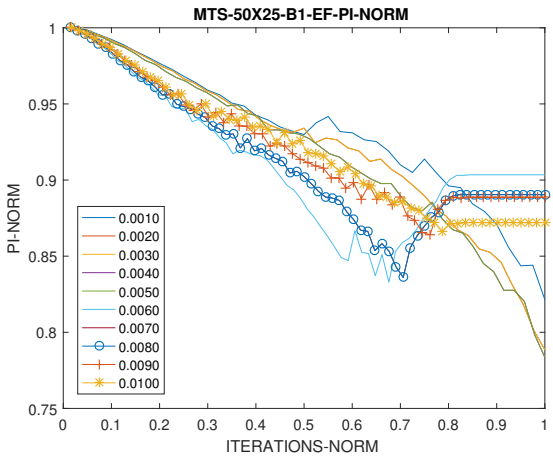
(a) MTS-50X25-B1-EF PI



(b) MTS-50X25-B1-EF PI normalized



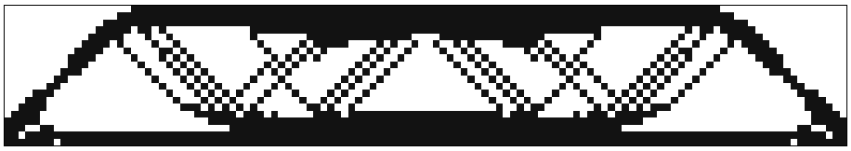
(c) MTS-50X25-B1-NF PI



(d) MTS-50X25-B1-NF PI normalized

Figure 8.2: Influence of the addition variable

(a) BS-60X10 E2



(b) BS-120X20 E2

(c) BS-180X40 E2

Figure 8.3: Mesh dependency for BS-120X20

8.3 Checkerboard pattern and mesh dependency

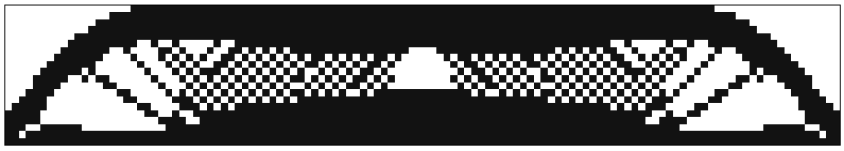
This test showed that all algorithms present checkerboard pattern in all benchmark cases and also mesh dependency. Remembering that BESO algorithm was implemented using filters, there was no need to do simulations with it. Some results for mesh dependency are presented in the Figure 8.3 for beam structure, Figure C.1 for Mitchell type structure, Figure C.2 for short cantilever and Figure C.3 for the two-bar frame cases. The complete list of results is presented in the Appendix C.

Results for the verification of checkerboard pattern are presented in the Figure 8.4 for the beam structure benchmark, in Figure C.4 for Mitchell type structure, Figure C.5 for short cantilever and Figure C.6 for the two-bar frame cases. The complete list of results is presented in the Appendix C.

8.4 Comparison between elemental and nodal filters and influence of the filters radius

Concerning the design for the filters, it is important to see the Figure 8.5. It was presented the visual result for each filter using the same radius applied to all benchmarks. It is clear that EF results in fewer holes with bigger diameters in the opposite of NF. At first, both have the same efficiency, D.1 and D.2. The major difference occurred when the final design was compared visually. The rest of the results are presented at Appendix D.

(a) BS-60X10 E3



(b) BS-120X20 E1

(c) BS-180X40 E2

Figure 8.4: Checkerboard pattern for BS-120X20

(a) BS-120X20-E1-EF 20

(b) BS-120X20-E1-NF 20

Figure 8.5: Filter radius results for BS-120X20

Table 8.3: Radius influence on the filters

		Filter radius [mm]			
		BS-120X20	MT-50X25	SC-48X30	TBF-25X60
ESO 1	EF	55	35	50	55
	NF	25	30	50	55
ESO 2	EF	35	25	70	55
	NF	30	35	55	55
ESO 3	EF	25	35	75	55
	NF	30	30	50	55
BESO	EF	35	30	45	95
	NF	40	45	90	90

8.5 influence of the radius on the filters results

Table 8.3 presents the results that had the lower PI for each algorithm, filter type, and benchmark. This evaluation was made by analyzing the Figure D.12, for example. Ir was repeated for all benchmarks in order to properly elaborate their results table. The rest of the results are presented at Appendix D.

8.6 3D application

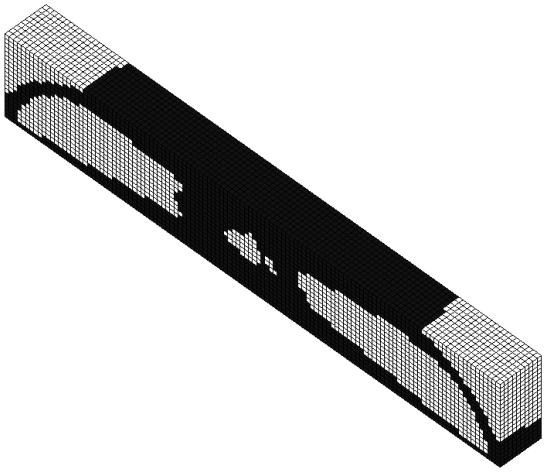
The evaluation of the efficiency of the algorithms applied to 3D models was satisfactory. The benchmarks used in the 2D studies are in essence 3D. They were not discretized in the thickness length. This test was able to confirm the applicability of them in 3D models. In order to simplify the time and number of simulations, this test has been done randomly mixing benchmark and parameters the way that all algorithms and filters were tested with one single model. Figure 8.6 presents on results for this study. The complete list of results is presented in the Table 8.4.

8.7 Efficiency of the evolutionary methods

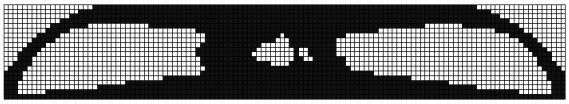
The last tests, the comparison between algorithms using optimized parameters, were realized following the Table F.1 presented in the Appendix E. Table 8.5 and 8.6 present the results for number of total iteration and total time for each case simulated. Among them all, the fastest simulation

Table 8.4: 3D Results

CASE	IT	TIME [hh:mm]
BS-120X20X10-E1-NF-25	39	02:15
MTS-50X25X10-E2-EF-25	39	01:41
SC-48X30X10-E3-NF-50	449	29:44
TBF-25X60X10-B1-EF-95	50	02:24



(a) BS-120X20X10 E1 NF 25 ISO view



(b) BS-120X20X10 E1 NF 25

Figure 8.6: 3D final design for BS-120X20X10

was the case SC-48X30-E2-SF, taking 1 minute and 04 seconds to calculate 39 iterations, while the slowest was the case BS-120X20-E2-EF-35, taking 54 minutes and 20 seconds for a same number of iterations

Considering the number of iterations, the simulation with fewer iterations was the case BS-120X20-E1-NF-25, performing 32 iterations in 5 minutes and 39 seconds and the larger number of iterations was 464, taking 11 minutes and 4 seconds for case MT-50X25-E3-SF. The respective values are represented in bold in the Tables 8.5 and 8.6.

Table 8.7 presents the results of the simulations with the best performance. The fastest simulation was accomplished using the ESO 1 algorithm and the slowest using the BESO. This table also shows that elemental filter had better results in comparison with the nodal. Figure 8.7 presents the final design for the optimum configurations for each benchmark. In figures of the BESO method result, the elements that have been reactivated by the algorithm are marked in black.

Table 8.5: Final comparison for BS-120X20 and MT-50X25

CASE	IT	TIME	CASE	IT	TIME
BS-120X20-E1-SF	39	00:01:26	MT-50X25-E1-SF	38	00:00:56
BS-120X20-E2-SF	39	00:01:27	MT-50X25-E2-SF	38	00:00:56
BS-120X20-E3-SF	446	00:16:13	MT-50X25-E3-SF	**464**	00:11:34
BS-120X20-E1-EF-55	30	00:01:09	MT-50X25-E1-EF-35	34	00:00:51
BS-120X20-E1-NF-25	**32**	00:05:39	MT-50X25-E1-NF-30	37	00:00:56
BS-120X20-E2-EF-35	39	**00:54:20**	MT-50X25-E2-EF-25	38	00:00:57
BS-120X20-E2-NF-30	39	00:01:27	MT-50X25-E2-NF-35	38	00:00:57
BS-120X20-E3-EF-25	372	00:13:54	MT-50X25-E3-EF-35	452	00:11:26
BS-120X20-E3-NF-30	420	00:13:54	MT-50X25-E3-NF-30	463	00:11:26
BS-120X20-B1-EF-35	66	00:02:29	MT-50X25-B1-EF-30	76	00:01:55
BS-120X20-B1-NF-40	94	00:03:33	MT-50X25-B1-NF-45	90	00:02:17

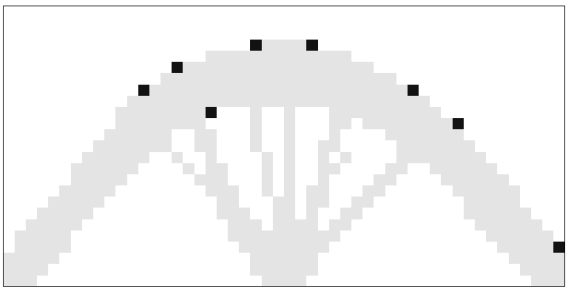
Table 8.6: Final comparison for SC-48X30 and TBF-25X60

CASE	IT	TIME	CASE	IT	TIME
SC-48X30-E1-SF	38	00:01:25	TBF-25X60-E1-SF	39	00:01:04
SC-48X30-E2-SF	38	00:01:25	TBF-25X60-E2-SF	39	**00:01:04**
SC-48X30-E3-SF	404	00:15:03	TBF-25X60-E3-SF	419	00:11:36
SC-48X30-E1-EF-50	36	00:01:21	TBF-25X60-E1-EF-55	38	00:01:04
SC-48X30-E1-NF-50	37	00:01:23	TBF-25X60-E1-NF-55	39	00:01:06
SC-48X30-E2-EF-70	38	00:01:26	TBF-25X60-E2-EF-55	39	00:01:05
SC-48X30-E2-NF-55	37	00:01:24	TBF-25X60-E2-NF-55	39	00:01:05
SC-48X30-E3-EF-75	365	00:13:45	TBF-25X60-E3-EF-55	407	00:11:27
SC-48X30-E3-NF-50	395	00:13:45	TBF-25X60-E3-NF-55	422	00:11:27
SC-48X30-B1-EF-45	71	00:02:41	TBF-25X60-B1-EF-95	74	00:02:06
SC-48X30-B1-NF-90	74	00:02:49	TBF-25X60-B1-NF-90	73	00:02:04

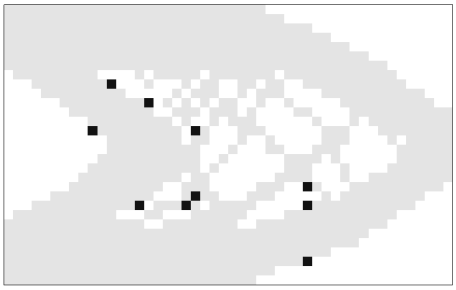
Table 8.7: Best results for each algorithm

CASE	IT	BREAK POINT	TIME
BS-120X20-E1-EF-55	30	25	00:01:09
MTS-50X25-B1-EF-30	76	44	00:01:55
SC-48X30-B1-EF-45	71	25	00:02:41
TBF-25X60-B1-EF-95	74	40	00:02:06

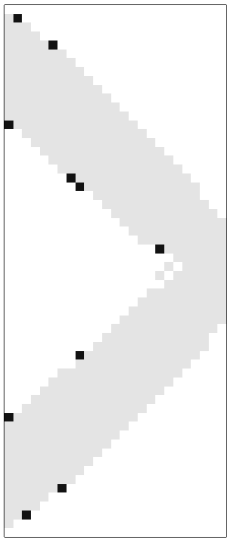
(a) BS-120X20-E1-EF-55



(b) MTS-50X25-B1-EF-30



(c) SC-48X30-B1-EF-45



(d) TBF-25X60-B1-EF-95

Figure 8.7: Final design for the best configuration responses

Chapter 9

Conclusion

In order to accomplish this work, a total of 760 simulations had been run, and 247 giga bytes of data had also been analyzed in the post-processing stage. After verifying the presence of problems like checkerboard pattern and mesh dependency, it has been proved that both appear in all benchmarks independently of the method used with it. The presence of these problems occurs in 100% of cases simulated without filters. Specifically for CP problem. Some cases just presented the initial formation of the CP like MTS and TBF benchmarks, Figure C.4 and C.6, while in the others it was quickly recognized. This test adequately explains the necessity of use of filters, either elemental or nodal, with the algorithms.

The α_{del_max} sensitivity test has shown that most of the models had similar values for ESO 1 and ESO 2 while for ESO 3 it had a little discrepancy. It may be explained by the use of α_{adj} that increases the number of elements deleted in some iterations by a small value. Nevertheless, SC was way off from the range, 20 times lower for ESO 1 and ESO 2 and, approximately 17 times higher for ESO 3. Since the majority had similar values and to minimize the total number of simulations that needed to be done, all cases were set to the same value for each algorithm. This approximation appears that appropriately worked fine for SC although the modification of its best α_{del_max}.

BESO had a particular sensitivity test to define the optimized α_{add}. However, it has the same initial structure of ESO 2, and this fact simplified the test. It should be done by crossing the α_{del_max} with α_{add}, to find the best pair of results for this algorithm indeed. However, it would have a high cost, an additional 90 simulations.

The analysis of elemental and nodal filters showed that the primary difference between them was the final design. EF provides solutions with fewer holes with bigger diameter sizes while the NF gives the opposite. It was expected that the total time of the NF should be higher than EF but, this was not proved in the final comparison. Generally, the total time was quite similar among them. Considering the way each one calculates the SN inside the filter area, NF uses more points and should need more time. The computational cost could define the use of one or the other. So, in a case that the speed is essential, EF should be the best choice. If the final design needs to be more visual clean or easily manufactured, NF should be used.

The 3D verification proved that all algorithms were able to be used in 3D simulations. Even with mesh with more than 10000 elements, the algorithm worked satisfactorily. Initially, ESO 3 is not well suited for this kind of simulation, since it needs in general three times more iterations than ESO 1 and 2. The longest time in all tests executed belongs precisely to ESO 3 with more than 29 hours of simulation. The better choice for 3D cases would be ESO 2.
The final comparison evaluation had some exciting outcomes. First, the use of PI was not enough to determinate the best results. At first, the lowest PI achieve was the criterion but, when ESO 1 had its lowest value, basically we arrived at the point that the structure collapsed. When another algorithm has lower PI, the breakpoint is still pretty much the same as ESO 1. It was not possible to precisely and quickly determine the best result for each benchmark using only PI. Here, would be a good practice check the von Mises stress, mean von Mises stress and the maximum displacement along with PI. Using these auxiliary criteria would be more comfortable to determinate the best case configuration. The code presented problems with symmetry. Initially, it was expected with BESO. The additional criteria are not ensuring that the number of reactivated elements are even because it has a bug on this block of the code and maybe it needs an additional control procedure to ensure the symmetric design response. However, the problem appeared with ESO 2 either. It may be caused by the difficulties of the code to delete an even number of elements on each iteration. α_{del} need to be monitored and checked to solve this bug.

After all the analysis, the algorithm ESO 1 with elemental filter had good results. It delivered good designs, and the symmetry was consistent, the total time of each simulation was lower in almost every case showed and EF uses less computational resources. Although, it seems that, as the fastest algorithm studied, the results may be unstable, since it deletes a higher number of elements at every iteration. When it delete a high amount of elements, and the structure had already achieved 50% of reduction or more, it became unstable swift and the structure collapse. The ESO 2, were more stable than ESO 1 since it deletes a constant number of elements along the process. It showed some problems with symmetry which resulted in a final design with lower quality, generally when used with EF. ESO 3 showed to be the most stable between all but, the time to achieve the desired volume reduction was at least ten times higher. It had good designs and thinking about this results, if the design needs to be optimized fast, ESO 1 is the choice, if the time is not essential, ESO 3. Paying always attention to the stability of the structure on ESO 1.

Chapter 10

Future Work

As future work on this subject, some ideas to solve some problems or improve the code are:

Implementation of other conditions as stop criterion for the optimization code (OC) . It was perceptible that using only the volume reduction as stop criteria for the OC was not enough. In order to upgrade the code and obtain better results the use of maximum displacement, maximum von Mises stress, minimum mean von Mises or 90% of the admissible tension of the material could be used. This change would approximate this OC to the real problems found in the engineering field. Any optimization would search for the lowest weight of the model under the specific tension. It means that any result would be efficient to the project parameters.

Rebuild the code to separate the SOLIDFEM from the OC SOLIDFEM was not built to works in a loop, so in the initial attempts to implement the OC, it was decided to modify it and use as a single software. In a future moment, after finishing this implementation work, re-built the OC as a new software separated from SOLIDFEM. To accomplish it, another step would be to adapt the SOLIDFEM to work with OC, read some variables to run the FEM simulation as a retro-compatible plugin.

Re-write the OC in order to solve some bugs as the asymmetric designs Review all the code to solve the problem that causes the OC crashing when compiled in release mode. Structurize the code to work in blocks and scripts, simplifying the time processing, use of memory and improving the overall efficiency.

Reduce the amount of RAM necessary to run the OC Since this code was an introduction to the optimization field, some variables and matrices were stored to help understanding how the code was working and finding bugs. In a later moment, this data stored could be written in text documents to reduce the total memory needed. Initially, some models needed more than 16Gb of RAM to be simulated. This improvement could reduce the time of simulation and allow refined meshes with more than 14000 elements to be used, most of them in 3D cases.

Study of the influence of evolutionary ratio adjust , α_{adj}, for the ESO 3 method. In this works, the value of α_{adj} was fixed in all cases to reduce the number of studies realized and total time to simulate everything. Specific to ESO 3, a new study comparing the α_{del_max} with α_{adj} should be done. Where for each α_{del_max} value a range of α_{adj} would be set and investigated to find the best parameters pair for this method.

Study of the α_{del_max} for each benchmark After the study of α_{del_max}, it is was noticed that the best value for this variable changed for each benchmark, specifically for the short cantilever case. Here an improvement in the pre-processing would solve the problem. Modifying the scripts that generate the data files with the case parameters and the one that generates the bat file that runs all the cases automatically.

Appendix A

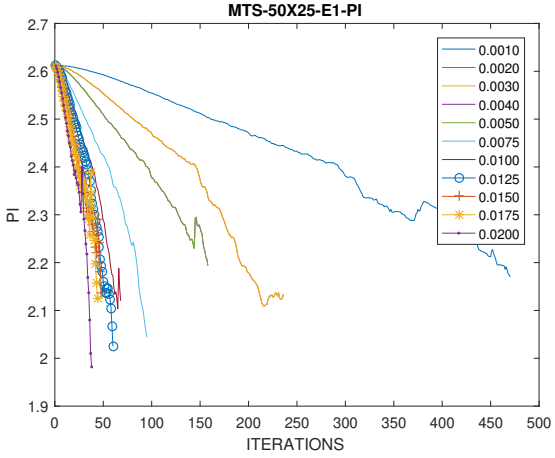
Eldelmax α_{del_max}

Table A.1: α_{del_max} cases abbreviation part 1

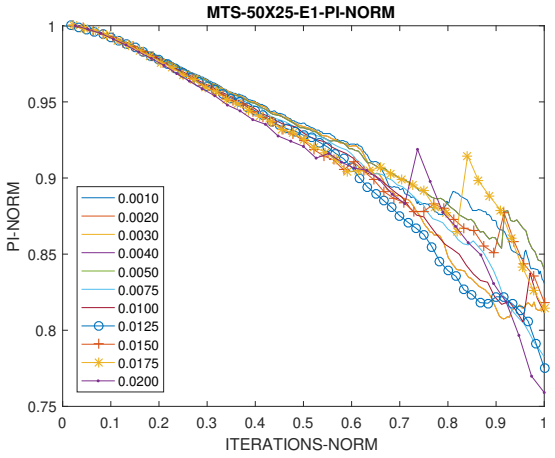
ELDELMAX	TBF-25X60	SC-48X30	MTS-50X25	BS-120X20
0.0010	TBF_25X60_E1_SF_0.001	SC_48X30_E1_SF_0.001	MTS_50X25_E1_SF_0.001	BS_120X20_E1_SF_0.001
0.0020	TBF_25X60_E1_SF_0.002	SC_48X30_E1_SF_0.002	MTS_50X25_E1_SF_0.002	BS_120X20_E1_SF_0.002
0.0030	TBF_25X60_E1_SF_0.003	SC_48X30_E1_SF_0.003	MTS_50X25_E1_SF_0.003	BS_120X20_E1_SF_0.003
0.0040	TBF_25X60_E1_SF_0.004	SC_48X30_E1_SF_0.004	MTS_50X25_E1_SF_0.004	BS_120X20_E1_SF_0.004
0.0050	TBF_25X60_E1_SF_0.005	SC_48X30_E1_SF_0.005	MTS_50X25_E1_SF_0.005	BS_120X20_E1_SF_0.005
0.0075	TBF_25X60_E1_SF_0.0075	SC_48X30_E1_SF_0.0075	MTS_50X25_E1_SF_0.0075	BS_120X20_E1_SF_0.0075
0.0100	TBF_25X60_E1_SF_0.01	SC_48X30_E1_SF_0.01	MTS_50X25_E1_SF_0.01	BS_120X20_E1_SF_0.01
0.0125	TBF_25X60_E1_SF_0.0125	SC_48X30_E1_SF_0.0125	MTS_50X25_E1_SF_0.0125	BS_120X20_E1_SF_0.0125
0.0150	TBF_25X60_E1_SF_0.015	SC_48X30_E1_SF_0.015	MTS_50X25_E1_SF_0.015	BS_120X20_E1_SF_0.015
0.0175	TBF_25X60_E1_SF_0.0175	SC_48X30_E1_SF_0.0175	MTS_50X25_E1_SF_0.0175	BS_120X20_E1_SF_0.0175
0.0200	TBF_25X60_E1_SF_0.02	SC_48X30_E1_SF_0.02	MTS_50X25_E1_SF_0.02	BS_120X20_E1_SF_0.02
0.0010	TBF_25X60_E2_SF_0.001	SC_48X30_E2_SF_0.001	MTS_50X25_E2_SF_0.001	BS_120X20_E2_SF_0.001
0.0020	TBF_25X60_E2_SF_0.002	SC_48X30_E2_SF_0.002	MTS_50X25_E2_SF_0.002	BS_120X20_E2_SF_0.002
0.0030	TBF_25X60_E2_SF_0.003	SC_48X30_E2_SF_0.003	MTS_50X25_E2_SF_0.003	BS_120X20_E2_SF_0.003
0.0040	TBF_25X60_E2_SF_0.004	SC_48X30_E2_SF_0.004	MTS_50X25_E2_SF_0.004	BS_120X20_E2_SF_0.004
0.0050	TBF_25X60_E2_SF_0.005	SC_48X30_E2_SF_0.005	MTS_50X25_E2_SF_0.005	BS_120X20_E2_SF_0.005
0.0075	TBF_25X60_E2_SF_0.0075	SC_48X30_E2_SF_0.0075	MTS_50X25_E2_SF_0.0075	BS_120X20_E2_SF_0.0075
0.0100	TBF_25X60_E2_SF_0.01	SC_48X30_E2_SF_0.01	MTS_50X25_E2_SF_0.01	BS_120X20_E2_SF_0.01
0.0125	TBF_25X60_E2_SF_0.0125	SC_48X30_E2_SF_0.0125	MTS_50X25_E2_SF_0.0125	BS_120X20_E2_SF_0.0125
0.0150	TBF_25X60_E2_SF_0.015	SC_48X30_E2_SF_0.015	MTS_50X25_E2_SF_0.015	BS_120X20_E2_SF_0.015
0.0175	TBF_25X60_E2_SF_0.0175	SC_48X30_E2_SF_0.0175	MTS_50X25_E2_SF_0.0175	BS_120X20_E2_SF_0.0175
0.0200	TBF_25X60_E2_SF_0.02	SC_48X30_E2_SF_0.02	MTS_50X25_E2_SF_0.02	BS_120X20_E2_SF_0.02

Table A.2: α_{del_max} cases abbreviation part 2

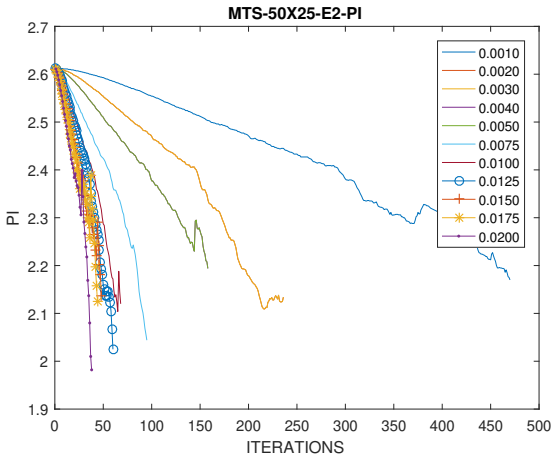
ELDELMAX	TBF-25X60	SC-48X30	MTS-50X25	BS-120X20
0.0010	TBF_25X60_E3_SF_0.001	SC_48X30_E3_SF_0.001	MTS_50X25_E3_SF_0.001	BS_120X20_E3_SF_0.001
0.0020	TBF_25X60_E3_SF_0.002	SC_48X30_E3_SF_0.002	MTS_50X25_E3_SF_0.002	BS_120X20_E3_SF_0.002
0.0030	TBF_25X60_E3_SF_0.003	SC_48X30_E3_SF_0.003	MTS_50X25_E3_SF_0.003	BS_120X20_E3_SF_0.003
0.0040	TBF_25X60_E3_SF_0.004	SC_48X30_E3_SF_0.004	MTS_50X25_E3_SF_0.004	BS_120X20_E3_SF_0.004
0.0050	TBF_25X60_E3_SF_0.005	SC_48X30_E3_SF_0.005	MTS_50X25_E3_SF_0.005	BS_120X20_E3_SF_0.005
0.0075	TBF_25X60_E3_SF_0.0075	SC_48X30_E3_SF_0.0075	MTS_50X25_E3_SF_0.0075	BS_120X20_E3_SF_0.0075
0.0100	TBF_25X60_E3_SF_0.01	SC_48X30_E3_SF_0.01	MTS_50X25_E3_SF_0.01	BS_120X20_E3_SF_0.01
0.0125	TBF_25X60_E3_SF_0.0125	SC_48X30_E3_SF_0.0125	MTS_50X25_E3_SF_0.0125	BS_120X20_E3_SF_0.0125
0.0150	TBF_25X60_E3_SF_0.015	SC_48X30_E3_SF_0.015	MTS_50X25_E3_SF_0.015	BS_120X20_E3_SF_0.015
0.0175	TBF_25X60_E3_SF_0.0175	SC_48X30_E3_SF_0.0175	MTS_50X25_E3_SF_0.0175	BS_120X20_E3_SF_0.0175
0.0200	TBF_25X60_E3_SF_0.02	SC_48X30_E3_SF_0.02	MTS_50X25_E3_SF_0.02	BS_120X20_E3_SF_0.02



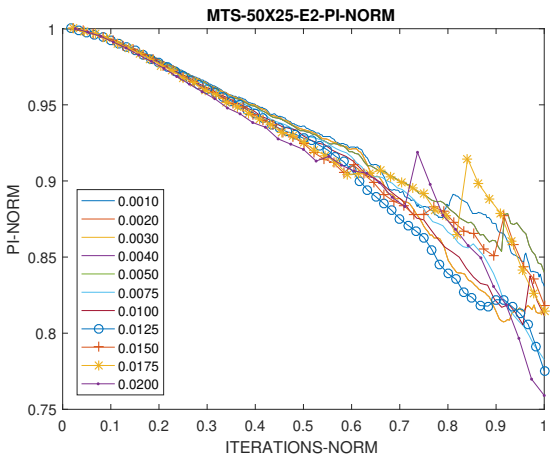
(a) MTS-50X25-E1 PI



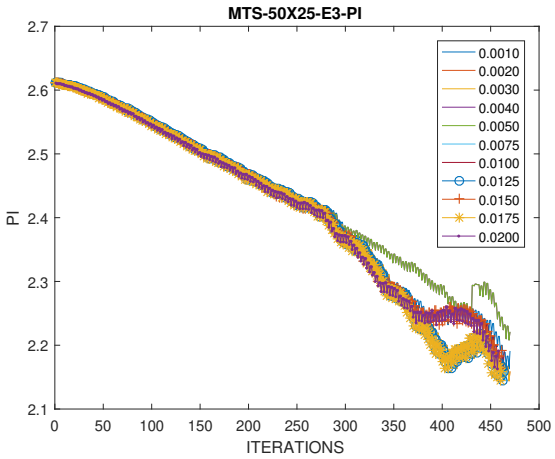
(b) MTS-50X25-E1 PI normalized



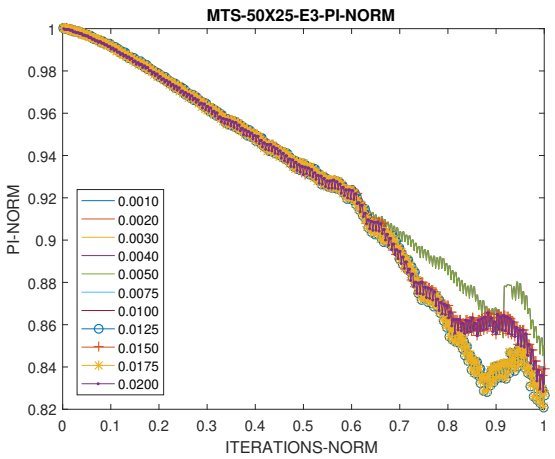
(c) MTS-50X25-E2 PI



(d) MTS-50X25-E2 PI normalized

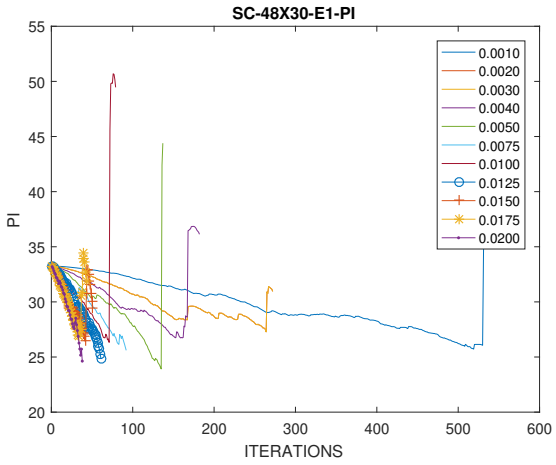


(e) MTS-50X25-E3 PI

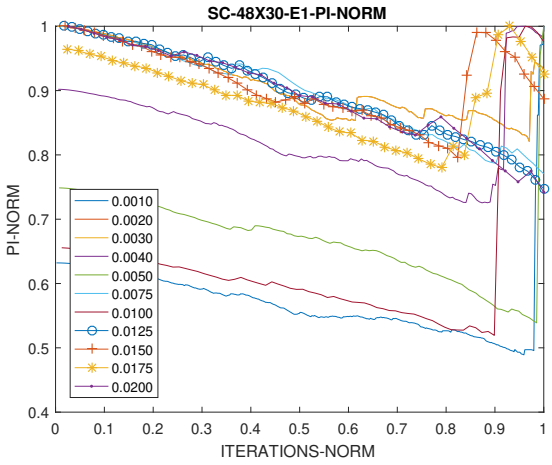


(f) MTS-50X25-E3 PI normalized

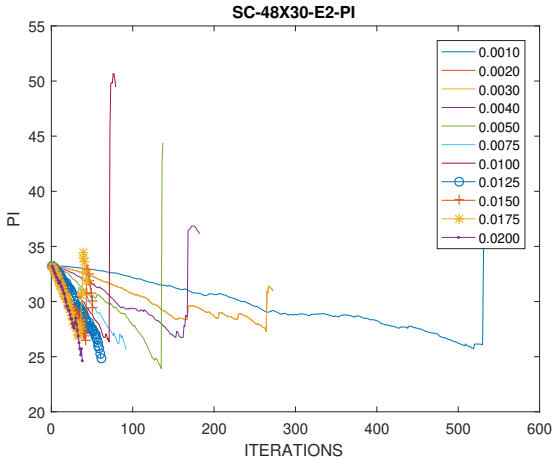
Figure A.1: α_{del_max} results for MTS-50X25



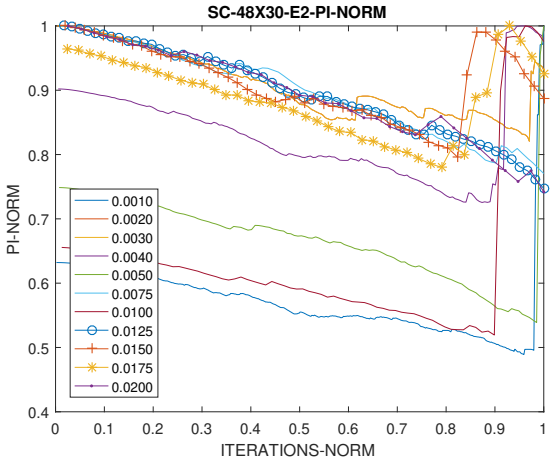
(a) SC-48X30-E1 PI



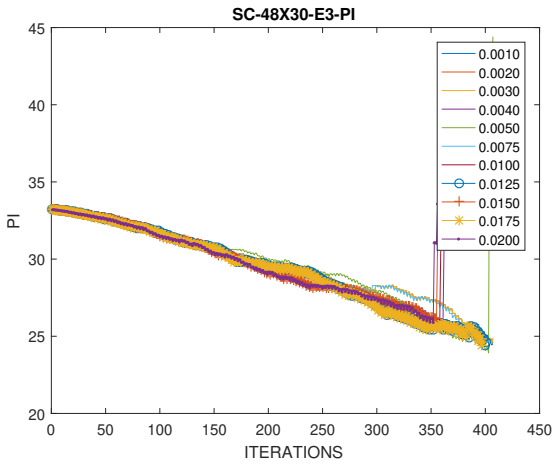
(b) SC-48X30-E1 PI normalized



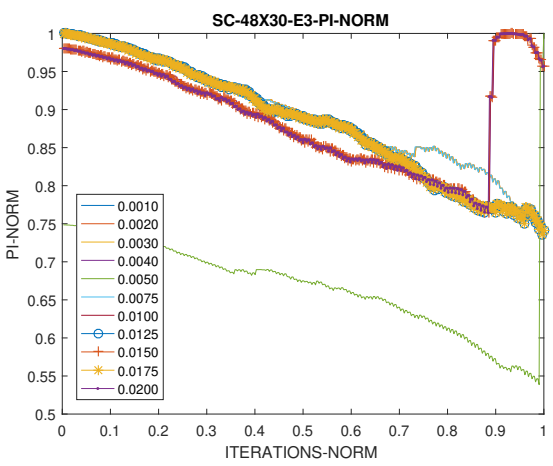
(c) SC-48X30-E2 PI



(d) SC-48X30-E2 PI normalized

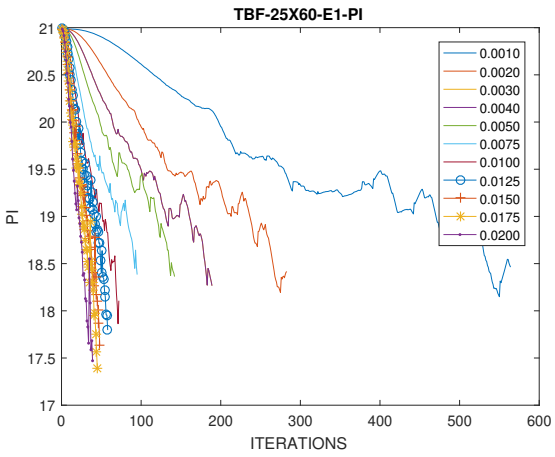


(e) SC-48X30-E3 PI

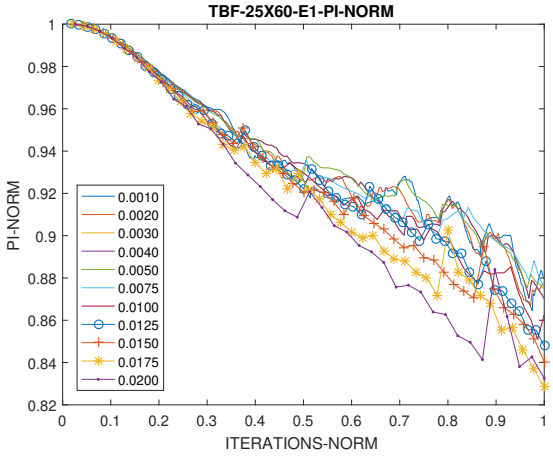


(f) SC-48X30-E3 PI normalized

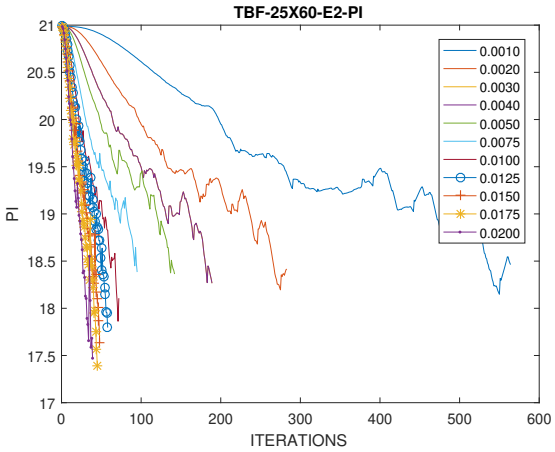
Figure A.2: α_{del_max} results for SC-48X30



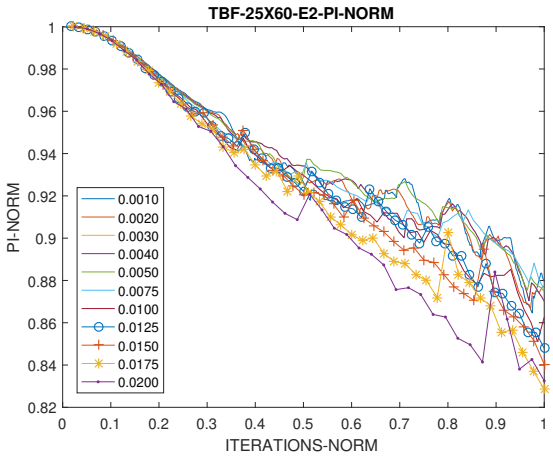
(a) TBF-25X60-E1 PI



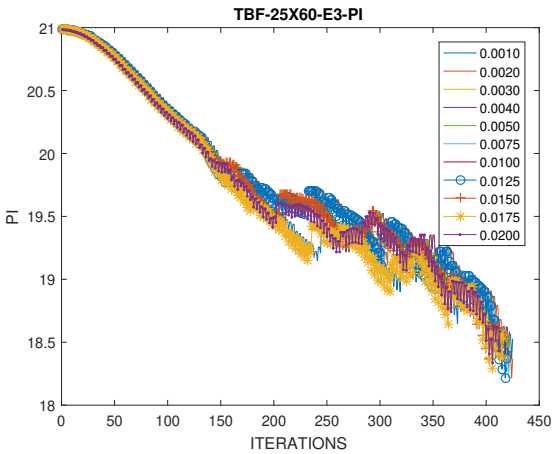
(b) TBF-25X60-E1 PI normalized



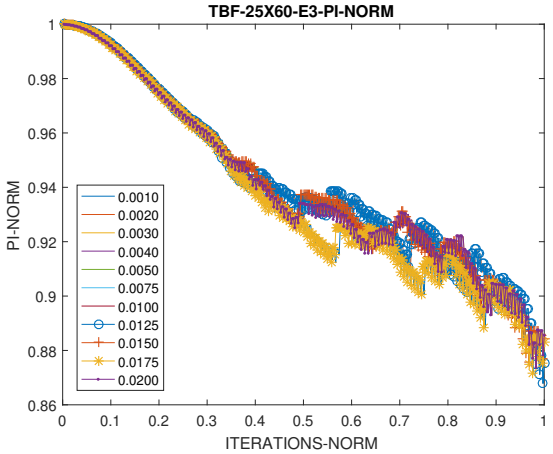
(c) TBF-25X60-E2 PI



(d) TBF-25X60-E2 PI normalized



(e) TBF-25X60-E3 PI



(f) TBF-25X60-E3 PI normalized

Figure A.3: α_{del_max} results for TBF-25X60

Appendix B

Beso addition

Table B.1: Beso addition cases abbreviation

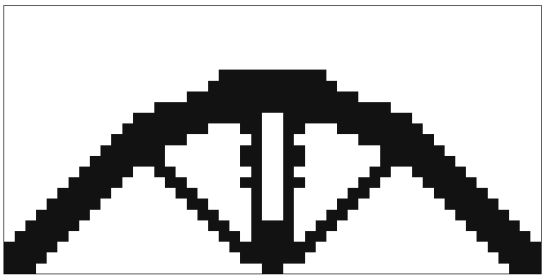
α_{add}	TBF_25X60	SC_48X30	MTS_50X25	BS_120X20
0.0005	TBF_25X60_BESO_NF_0.0005	SC_48X30_BESO_NF_0.0005	MTS_50X25_BESO_NF_0.0005	BS_120X20_BESO_NF_0.0005
0.0010	TBF_25X60_BESO_NF_0.001	SC_48X30_BESO_NF_0.001	MTS_50X25_BESO_NF_0.001	BS_120X20_BESO_NF_0.001
0.0015	TBF_25X60_BESO_NF_0.0015	SC_48X30_BESO_NF_0.0015	MTS_50X25_BESO_NF_0.0015	BS_120X20_BESO_NF_0.0015
0.0020	TBF_25X60_BESO_NF_0.002	SC_48X30_BESO_NF_0.002	MTS_50X25_BESO_NF_0.002	BS_120X20_BESO_NF_0.002
0.0025	TBF_25X60_BESO_NF_0.0025	SC_48X30_BESO_NF_0.0025	MTS_50X25_BESO_NF_0.0025	BS_120X20_BESO_NF_0.0025
0.0030	TBF_25X60_BESO_NF_0.003	SC_48X30_BESO_NF_0.003	MTS_50X25_BESO_NF_0.003	BS_120X20_BESO_NF_0.003
0.0035	TBF_25X60_BESO_NF_0.0035	SC_48X30_BESO_NF_0.0035	MTS_50X25_BESO_NF_0.0035	BS_120X20_BESO_NF_0.0035
0.0040	TBF_25X60_BESO_NF_0.004	SC_48X30_BESO_NF_0.004	MTS_50X25_BESO_NF_0.004	BS_120X20_BESO_NF_0.004
0.0045	TBF_25X60_BESO_NF_0.0045	SC_48X30_BESO_NF_0.0045	MTS_50X25_BESO_NF_0.0045	BS_120X20_BESO_NF_0.0045
0.0050	TBF_25X60_BESO_NF_0.005	SC_48X30_BESO_NF_0.005	MTS_50X25_BESO_NF_0.005	BS_120X20_BESO_NF_0.005

Appendix C

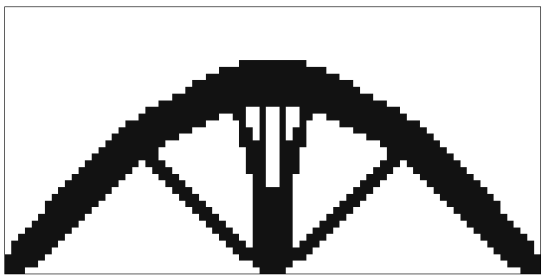
Checkerboard pattern and mesh dependency

Table C.1: Checkerboard pattern and mesh dependency cases abbreviation

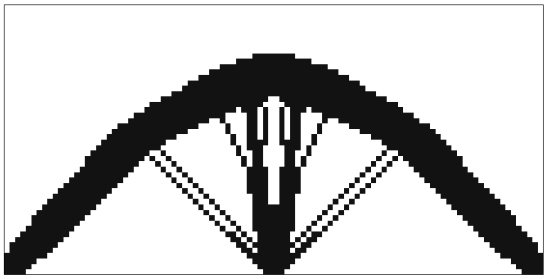
BS	MTS	SC	TBF
TBF_25X60_E1_SF	TBF_25X60_E1_SF	TBF_25X60_E1_SF	TBF_25X60_E1_SF
TBF_25X60_E2_SF	TBF_25X60_E2_SF	TBF_25X60_E2_SF	TBF_25X60_E2_SF
TBF_25X60_E3_SF	TBF_25X60_E3_SF	TBF_25X60_E3_SF	TBF_25X60_E3_SF
TBF_50X120_E1_SF	TBF_50X120_E1_SF	TBF_50X120_E1_SF	TBF_50X120_E1_SF
TBF_50X120_E2_SF	TBF_50X120_E2_SF	TBF_50X120_E2_SF	TBF_50X120_E2_SF
TBF_50X120_E3_SF	TBF_50X120_E3_SF	TBF_50X120_E3_SF	TBF_50X120_E3_SF
TBF_75X80_E1_SF	TBF_75X80_E1_SF	TBF_75X80_E1_SF	TBF_75X80_E1_SF
TBF_75X80_E2_SF	TBF_75X80_E2_SF	TBF_75X80_E2_SF	TBF_75X80_E2_SF
TBF_75X80_E3_SF	TBF_75X80_E3_SF	TBF_75X80_E3_SF	TBF_75X80_E3_SF



(a) MTS-50X25 E3

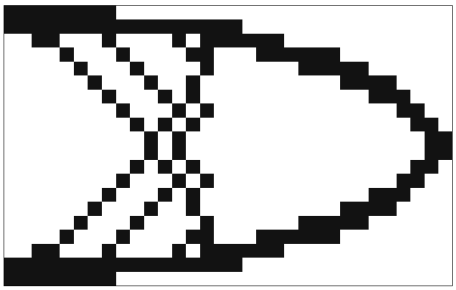


(b) MTS-80X40 E3

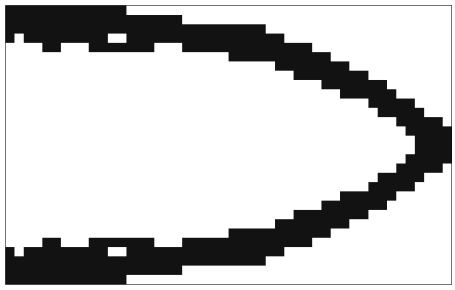


(c) MTS-100X50 E3

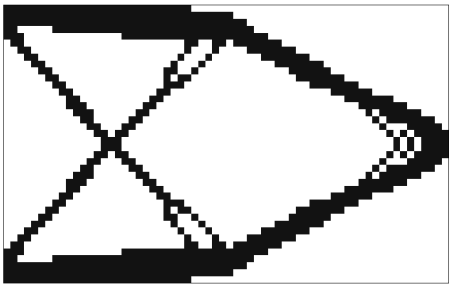
Figure C.1: Mesh dependency for E3 MTS-25X60



(a) SC-32X20 E1

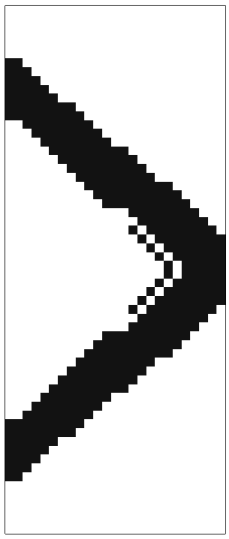


(b) SC-48X30 E1

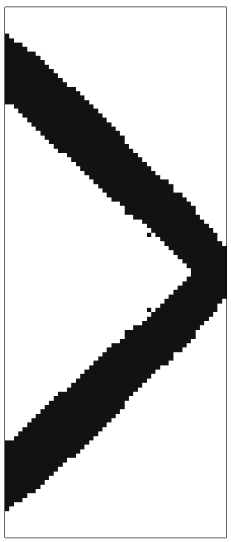


(c) SC-64X400 E1

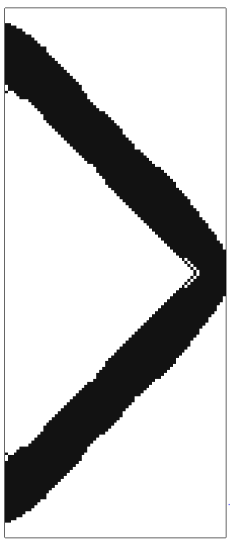
Figure C.2: Mesh dependency for E1 SC-48X30



(a) TBF-25X60 E2

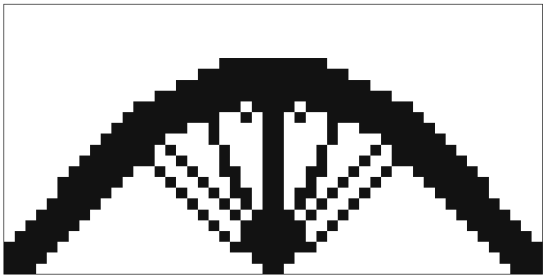


(b) TBF-50X120 E2

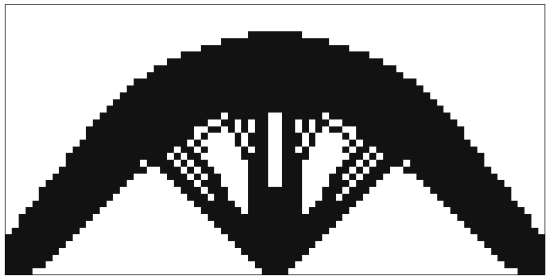


(c) TBF-75X180 E2

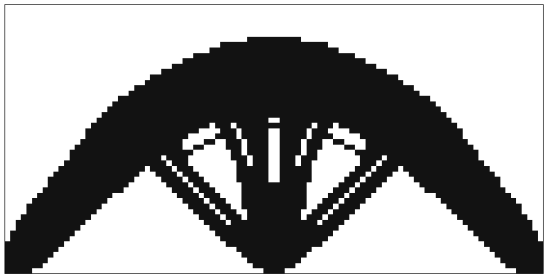
Figure C.3: Mesh dependency for E2 TBF-25X60



(a) MTS-50X25 E1

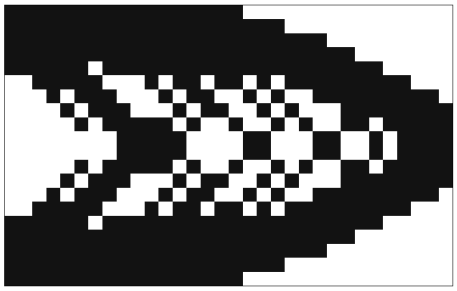


(b) MTS-80X40 E2

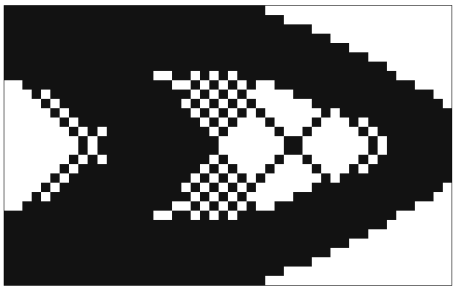


(c) MTS-100X50 E3

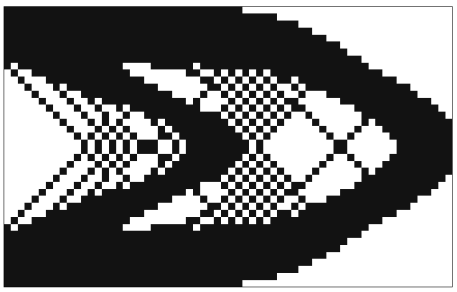
Figure C.4: Checkerboard pattern for MTS-25X60



(a) SC-32X20 E1

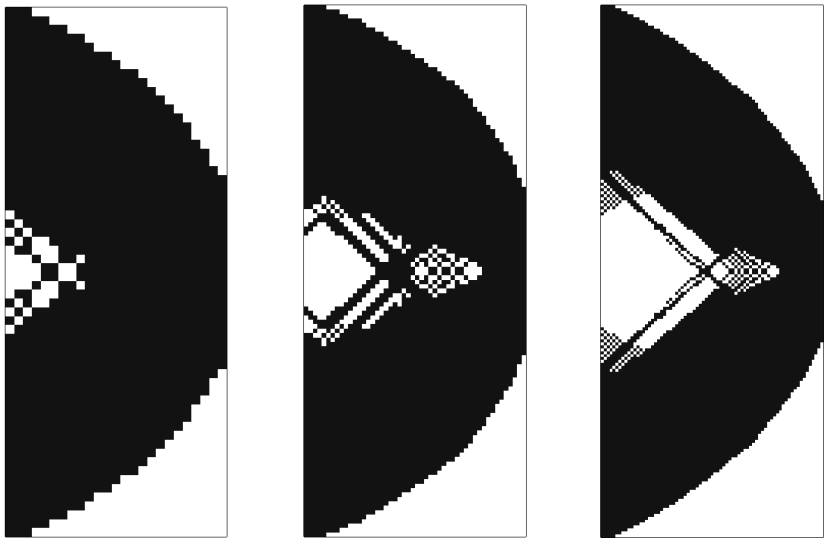


(b) SC-48X30 E2



(c) SC-64X400 E3

Figure C.5: Checkerboard pattern for SC-48X30



(a) TBF-25X60 E2 (b) TBF-50X120 E2 (c) TBF-75X180 E2

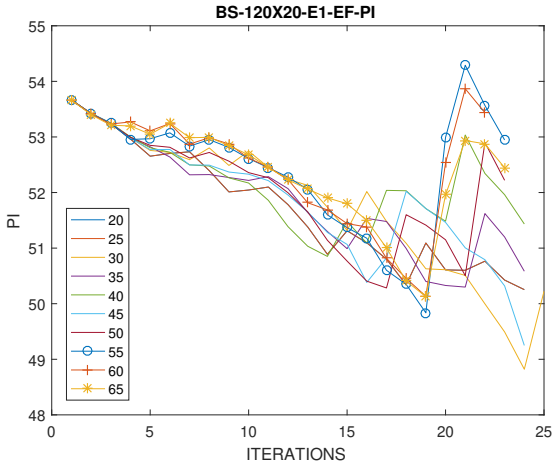
Figure C.6: Checkerboard pattern for TBF-25X60

Appendix D

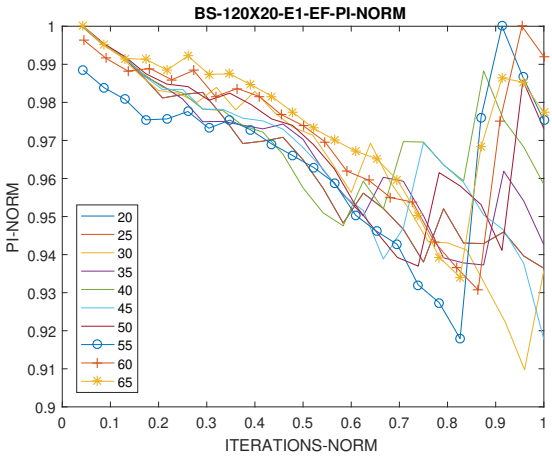
Elemental and nodal filter

Table D.1: Elemental and nodal filter cases abbreviation

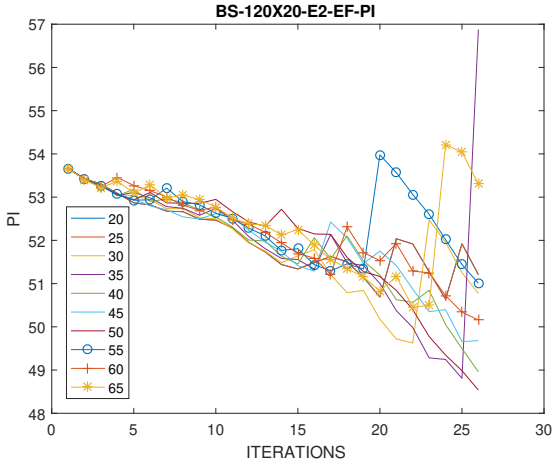
BS_120X20		MTS_50X25		SC_48X30		TBF_25X60	
FILTER		FILTER		FILTER		FILTER	
ELEMENTAL	NODAL	ELEMENTAL	NODAL	ELEMENTAL	NODAL	ELEMENTAL	NODAL
BS_120X20_E1_EF_20	BS_120X20_E1_NF_20	MTS_50X25_E1_EF_20	MTS_50X25_E1_NF_20	SC_48X30_E1_EF_35	SC_48X30_E1_NF_35	TBF_25X60_E1_EF_40	TBF_25X60_E1_NF_40
BS_120X20_E1_EF_25	BS_120X20_E1_NF_25	MTS_50X25_E1_EF_25	MTS_50X25_E1_NF_25	SC_48X30_E1_EF_40	SC_48X30_E1_NF_40	TBF_25X60_E1_EF_45	TBF_25X60_E1_NF_45
BS_120X20_E1_EF_30	BS_120X20_E1_NF_30	MTS_50X25_E1_EF_30	MTS_50X25_E1_NF_30	SC_48X30_E1_EF_45	SC_48X30_E1_NF_45	TBF_25X60_E1_EF_50	TBF_25X60_E1_NF_50
BS_120X20_E1_EF_35	BS_120X20_E1_NF_35	MTS_50X25_E1_EF_35	MTS_50X25_E1_NF_35	SC_48X30_E1_EF_50	SC_48X30_E1_NF_50	TBF_25X60_E1_EF_55	TBF_25X60_E1_NF_55
BS_120X20_E1_EF_40	BS_120X20_E1_NF_40	MTS_50X25_E1_EF_40	MTS_50X25_E1_NF_40	SC_48X30_E1_EF_55	SC_48X30_E1_NF_55	TBF_25X60_E1_EF_60	TBF_25X60_E1_NF_60
BS_120X20_E1_EF_45	BS_120X20_E1_NF_45	MTS_50X25_E1_EF_45	MTS_50X25_E1_NF_45	SC_48X30_E1_EF_60	SC_48X30_E1_NF_60	TBF_25X60_E1_EF_65	TBF_25X60_E1_NF_65
BS_120X20_E1_EF_50	BS_120X20_E1_NF_50	MTS_50X25_E1_EF_50	MTS_50X25_E1_NF_50	SC_48X30_E1_EF_65	SC_48X30_E1_NF_65	TBF_25X60_E1_EF_70	TBF_25X60_E1_NF_70
BS_120X20_E1_EF_55	BS_120X20_E1_NF_55	MTS_50X25_E1_EF_55	MTS_50X25_E1_NF_55	SC_48X30_E1_EF_70	SC_48X30_E1_NF_70	TBF_25X60_E1_EF_75	TBF_25X60_E1_NF_75
BS_120X20_E1_EF_60	BS_120X20_E1_NF_60	MTS_50X25_E1_EF_60	MTS_50X25_E1_NF_60	SC_48X30_E1_EF_75	SC_48X30_E1_NF_75	TBF_25X60_E1_EF_80	TBF_25X60_E1_NF_80
BS_120X20_E1_EF_65	BS_120X20_E1_NF_65	MTS_50X25_E1_EF_65	MTS_50X25_E1_NF_65	SC_48X30_E1_EF_80	SC_48X30_E1_NF_80	TBF_25X60_E1_EF_85	TBF_25X60_E1_NF_85
				SC_48X30_E1_EF_85	SC_48X30_E1_NF_85	TBF_25X60_E1_EF_90	TBF_25X60_E1_NF_90
				SC_48X30_E1_EF_90	SC_48X30_E1_NF_90	TBF_25X60_E1_EF_95	TBF_25X60_E1_NF_95
				SC_48X30_E1_EF_95	SC_48X30_E1_NF_95	TBF_25X60_E1_EF_100	TBF_25X60_E1_NF_100
				SC_48X30_E1_EF_100	SC_48X30_E1_NF_100	TBF_25X60_E1_EF_105	TBF_25X60_E1_NF_105
				SC_48X30_E1_EF_105	SC_48X30_E1_NF_105	TBF_25X60_E1_EF_110	TBF_25X60_E1_NF_110
						TBF_25X60_E1_EF_115	TBF_25X60_E1_NF_115
						TBF_25X60_E1_EF_120	TBF_25X60_E1_NF_120
BS_120X20_E2_EF_20	BS_120X20_E2_NF_20	MTS_50X25_E2_EF_20	MTS_50X25_E2_NF_20	SC_48X30_E2_EF_35	SC_48X30_E2_NF_35	TBF_25X60_E2_EF_40	TBF_25X60_E2_NF_40
BS_120X20_E2_EF_25	BS_120X20_E2_NF_25	MTS_50X25_E2_EF_25	MTS_50X25_E2_NF_25	SC_48X30_E2_EF_40	SC_48X30_E2_NF_40	TBF_25X60_E2_EF_45	TBF_25X60_E2_NF_45
BS_120X20_E2_EF_30	BS_120X20_E2_NF_30	MTS_50X25_E2_EF_30	MTS_50X25_E2_NF_30	SC_48X30_E2_EF_45	SC_48X30_E2_NF_45	TBF_25X60_E2_EF_50	TBF_25X60_E2_NF_50
BS_120X20_E2_EF_35	BS_120X20_E2_NF_35	MTS_50X25_E2_EF_35	MTS_50X25_E2_NF_35	SC_48X30_E2_EF_50	SC_48X30_E2_NF_50	TBF_25X60_E2_EF_55	TBF_25X60_E2_NF_55
BS_120X20_E2_EF_40	BS_120X20_E2_NF_40	MTS_50X25_E2_EF_40	MTS_50X25_E2_NF_40	SC_48X30_E2_EF_55	SC_48X30_E2_NF_55	TBF_25X60_E2_EF_60	TBF_25X60_E2_NF_60
BS_120X20_E2_EF_45	BS_120X20_E2_NF_45	MTS_50X25_E2_EF_45	MTS_50X25_E2_NF_45	SC_48X30_E2_EF_60	SC_48X30_E2_NF_60	TBF_25X60_E2_EF_65	TBF_25X60_E2_NF_65
BS_120X20_E2_EF_50	BS_120X20_E2_NF_50	MTS_50X25_E2_EF_50	MTS_50X25_E2_NF_50	SC_48X30_E2_EF_65	SC_48X30_E2_NF_65	TBF_25X60_E2_EF_70	TBF_25X60_E2_NF_70
BS_120X20_E2_EF_55	BS_120X20_E2_NF_55	MTS_50X25_E2_EF_55	MTS_50X25_E2_NF_55	SC_48X30_E2_EF_70	SC_48X30_E2_NF_70	TBF_25X60_E2_EF_75	TBF_25X60_E2_NF_75
BS_120X20_E2_EF_60	BS_120X20_E2_NF_60	MTS_50X25_E2_EF_60	MTS_50X25_E2_NF_60	SC_48X30_E2_EF_75	SC_48X30_E2_NF_75	TBF_25X60_E2_EF_80	TBF_25X60_E2_NF_80
BS_120X20_E2_EF_65	BS_120X20_E2_NF_65	MTS_50X25_E2_EF_65	MTS_50X25_E2_NF_65	SC_48X30_E2_EF_80	SC_48X30_E2_NF_80	TBF_25X60_E2_EF_85	TBF_25X60_E2_NF_85
				SC_48X30_E2_EF_85	SC_48X30_E2_NF_85	TBF_25X60_E2_EF_90	TBF_25X60_E2_NF_90
				SC_48X30_E2_EF_90	SC_48X30_E2_NF_90	TBF_25X60_E2_EF_95	TBF_25X60_E2_NF_95
				SC_48X30_E2_EF_95	SC_48X30_E2_NF_95	TBF_25X60_E2_EF_100	TBF_25X60_E2_NF_100
				SC_48X30_E2_EF_100	SC_48X30_E2_NF_100	TBF_25X60_E2_EF_105	TBF_25X60_E2_NF_105
				SC_48X30_E2_EF_105	SC_48X30_E2_NF_105	TBF_25X60_E2_EF_110	TBF_25X60_E2_NF_110
						TBF_25X60_E2_EF_115	TBF_25X60_E2_NF_115
						TBF_25X60_E2_EF_120	TBF_25X60_E2_NF_120
BS_120X20_E3_EF_20	BS_120X20_E3_NF_20	MTS_50X25_E3_EF_20	MTS_50X25_E3_NF_20	SC_48X30_E3_EF_35	SC_48X30_E3_NF_35	TBF_25X60_E3_EF_40	TBF_25X60_E3_NF_40
BS_120X20_E3_EF_25	BS_120X20_E3_NF_25	MTS_50X25_E3_EF_25	MTS_50X25_E3_NF_25	SC_48X30_E3_EF_40	SC_48X30_E3_NF_40	TBF_25X60_E3_EF_45	TBF_25X60_E3_NF_45
BS_120X20_E3_EF_30	BS_120X20_E3_NF_30	MTS_50X25_E3_EF_30	MTS_50X25_E3_NF_30	SC_48X30_E3_EF_45	SC_48X30_E3_NF_45	TBF_25X60_E3_EF_50	TBF_25X60_E3_NF_50
BS_120X20_E3_EF_35	BS_120X20_E3_NF_35	MTS_50X25_E3_EF_35	MTS_50X25_E3_NF_35	SC_48X30_E3_EF_50	SC_48X30_E3_NF_50	TBF_25X60_E3_EF_55	TBF_25X60_E3_NF_55
BS_120X20_E3_EF_40	BS_120X20_E3_NF_40	MTS_50X25_E3_EF_40	MTS_50X25_E3_NF_40	SC_48X30_E3_EF_55	SC_48X30_E3_NF_55	TBF_25X60_E3_EF_60	TBF_25X60_E3_NF_60
BS_120X20_E3_EF_45	BS_120X20_E3_NF_45	MTS_50X25_E3_EF_45	MTS_50X25_E3_NF_45	SC_48X30_E3_EF_60	SC_48X30_E3_NF_60	TBF_25X60_E3_EF_65	TBF_25X60_E3_NF_65
BS_120X20_E3_EF_50	BS_120X20_E3_NF_50	MTS_50X25_E3_EF_50	MTS_50X25_E3_NF_50	SC_48X30_E3_EF_65	SC_48X30_E3_NF_65	TBF_25X60_E3_EF_70	TBF_25X60_E3_NF_70
BS_120X20_E3_EF_55	BS_120X20_E3_NF_55	MTS_50X25_E3_EF_55	MTS_50X25_E3_NF_55	SC_48X30_E3_EF_70	SC_48X30_E3_NF_70	TBF_25X60_E3_EF_75	TBF_25X60_E3_NF_75
BS_120X20_E3_EF_60	BS_120X20_E3_NF_60	MTS_50X25_E3_EF_60	MTS_50X25_E3_NF_60	SC_48X30_E3_EF_75	SC_48X30_E3_NF_75	TBF_25X60_E3_EF_80	TBF_25X60_E3_NF_80
BS_120X20_E3_EF_65	BS_120X20_E3_NF_65	MTS_50X25_E3_EF_65	MTS_50X25_E3_NF_65	SC_48X30_E3_EF_80	SC_48X30_E3_NF_80	TBF_25X60_E3_EF_85	TBF_25X60_E3_NF_85
				SC_48X30_E3_EF_85	SC_48X30_E3_NF_85	TBF_25X60_E3_EF_90	TBF_25X60_E3_NF_90
				SC_48X30_E3_EF_90	SC_48X30_E3_NF_90	TBF_25X60_E3_EF_95	TBF_25X60_E3_NF_95
				SC_48X30_E3_EF_95	SC_48X30_E3_NF_95	TBF_25X60_E3_EF_100	TBF_25X60_E3_NF_100
				SC_48X30_E3_EF_100	SC_48X30_E3_NF_100	TBF_25X60_E3_EF_105	TBF_25X60_E3_NF_105
				SC_48X30_E3_EF_105	SC_48X30_E3_NF_105	TBF_25X60_E3_EF_110	TBF_25X60_E3_NF_110
						TBF_25X60_E3_EF_115	TBF_25X60_E3_NF_115
						TBF_25X60_E3_EF_120	TBF_25X60_E3_NF_120



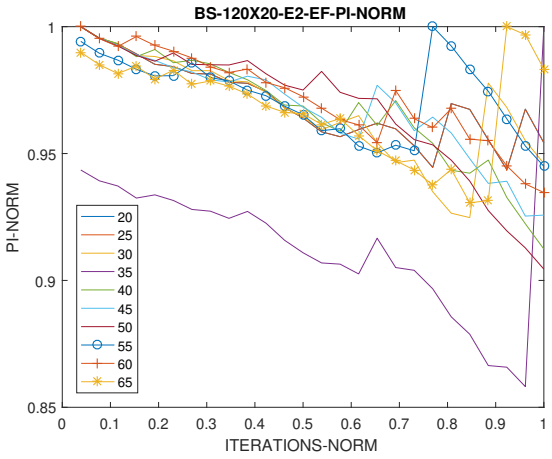
(a) BS-120X20-E1-EF PI



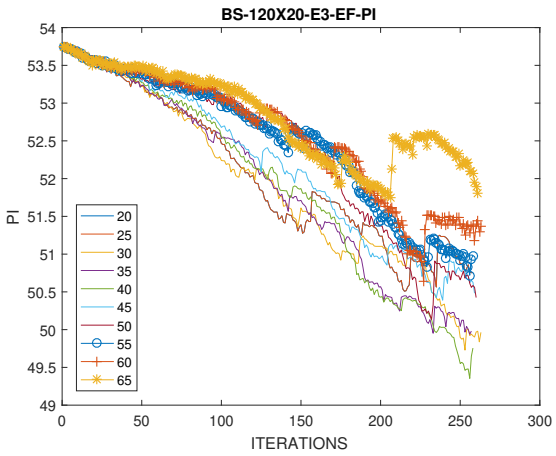
(b) BS-120X20-E1-EF PI normalized



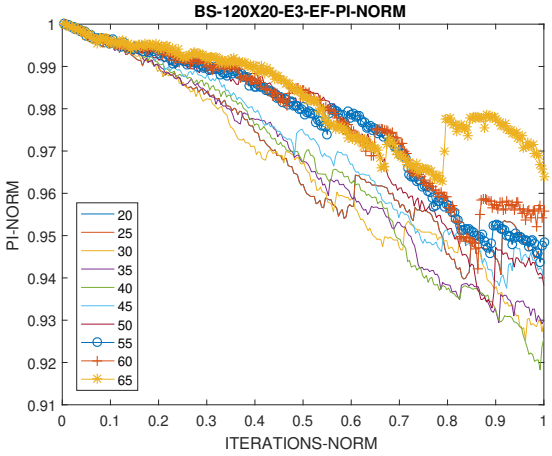
(c) BS-120X20-E2-EF PI



(d) BS-120X20-E2-EF PI normalized

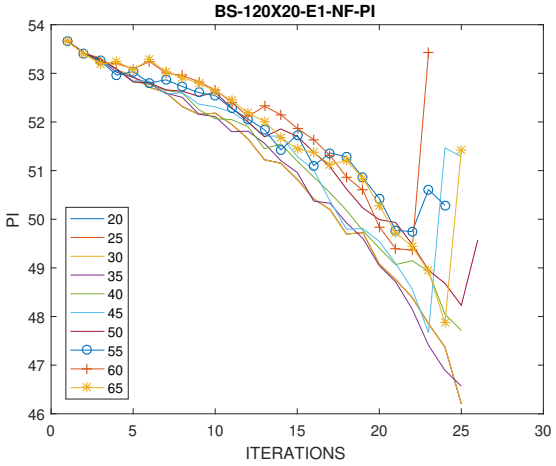


(e) BS-120X20-E3-EF PI

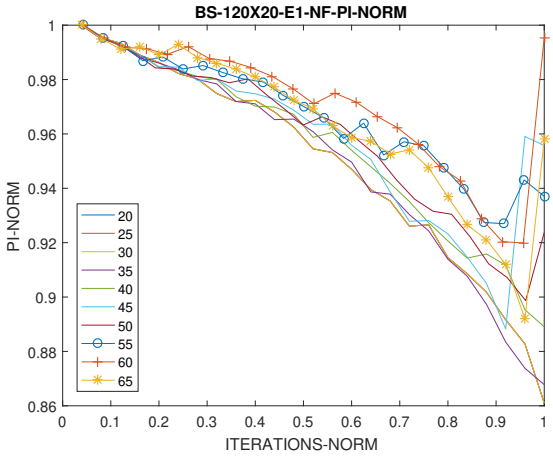


(f) BS-120X20-E3-EF PI normalized

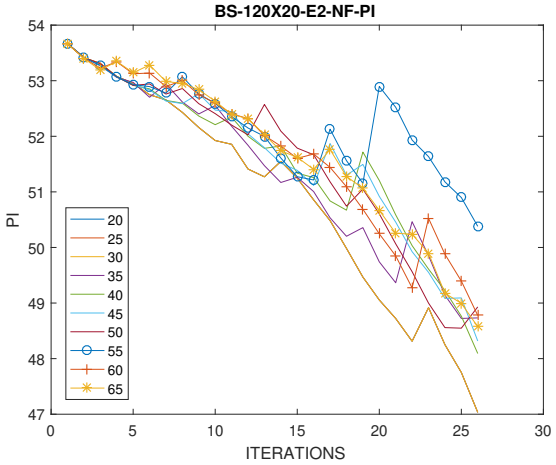
Figure D.1: Elemental filter radius results for BS-120X20



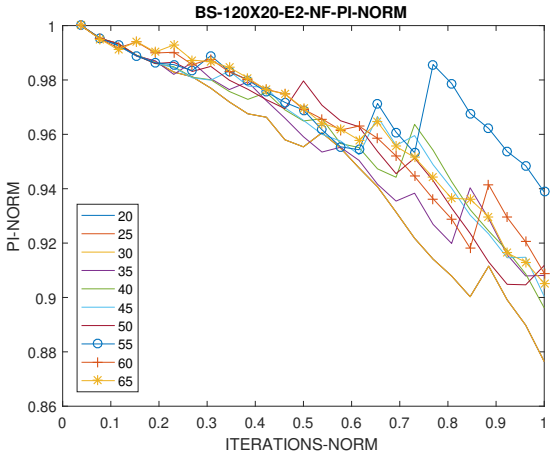
(a) BS-120X20-E1-NF PI



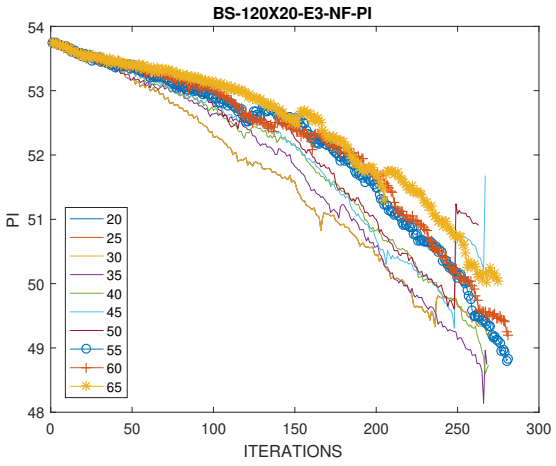
(b) BS-120X20-E1-NF PI normalized



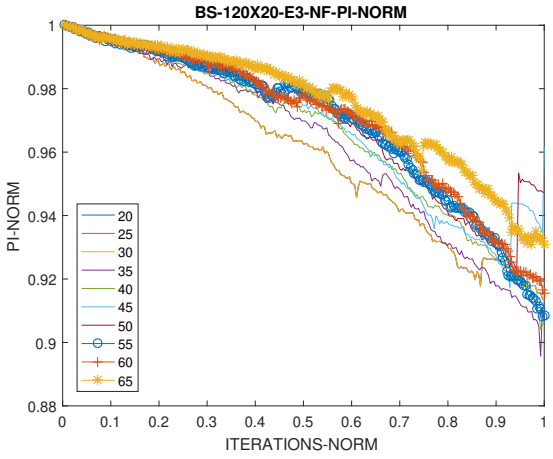
(c) BS-120X20-E2-NF PI



(d) BS-120X20-E2-NF PI normalized

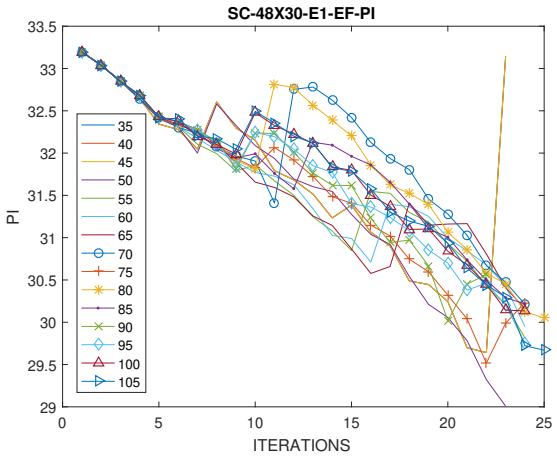


(e) BS-120X20-E3-NF PI

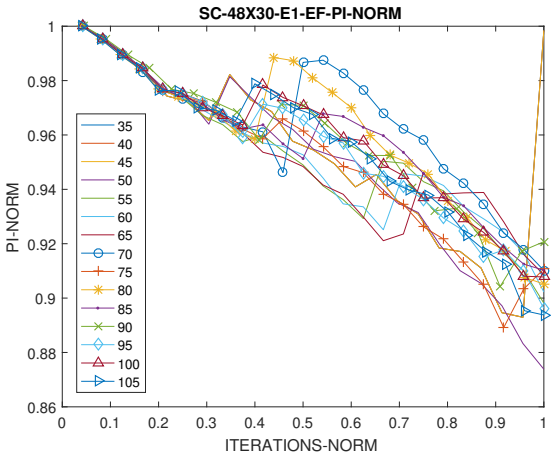


(f) BS-120X20-E3-NF PI normalized

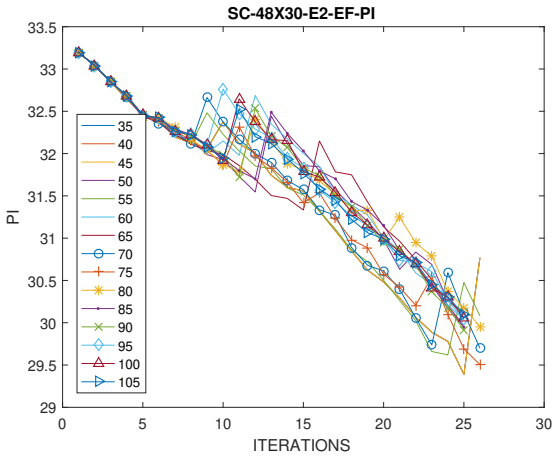
Figure D.2: Nodal filter radius results for BS-120X20



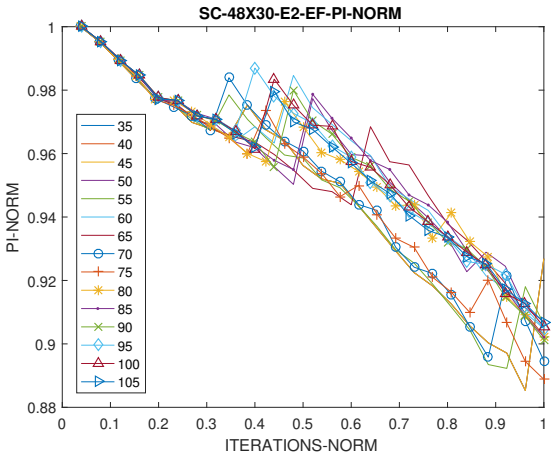
(a) SC-48X30-E1-EF PI



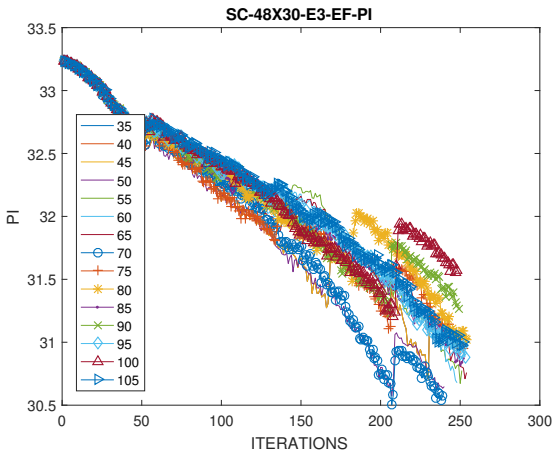
(b) SC-48X30-E1-EF PI normalized



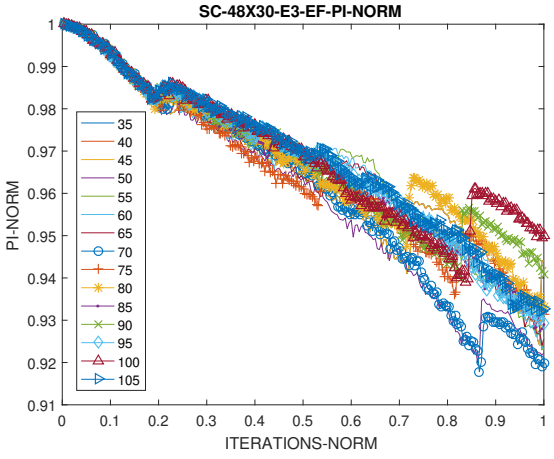
(c) SC-48X30-E2-EF PI



(d) SC-48X30-E2-EF PI normalized

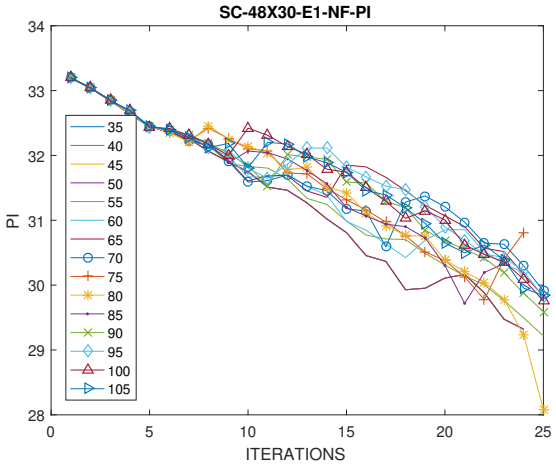


(e) SC-48X30-E3-EF PI

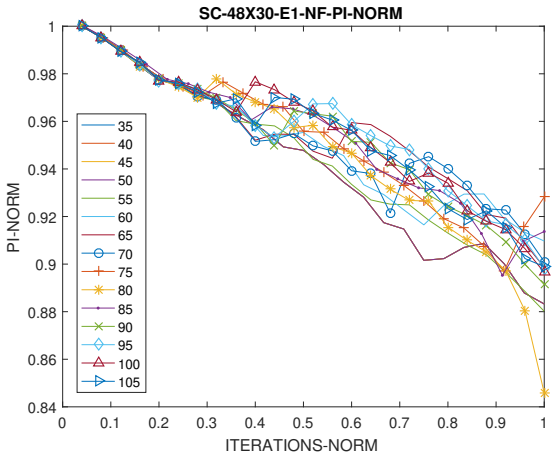


(f) SC-48X30-E3-EF PI normalized

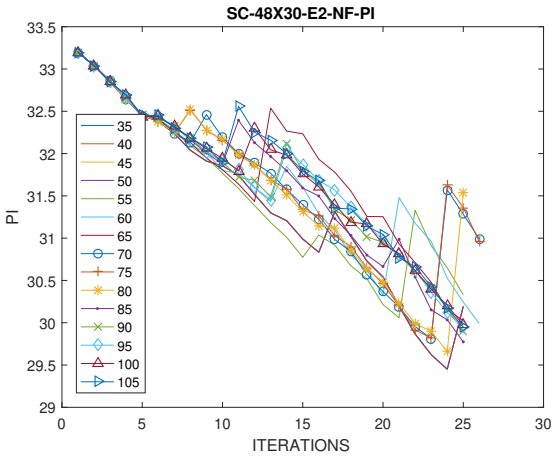
Figure D.3: Elemental filter radius results for SC-48X30



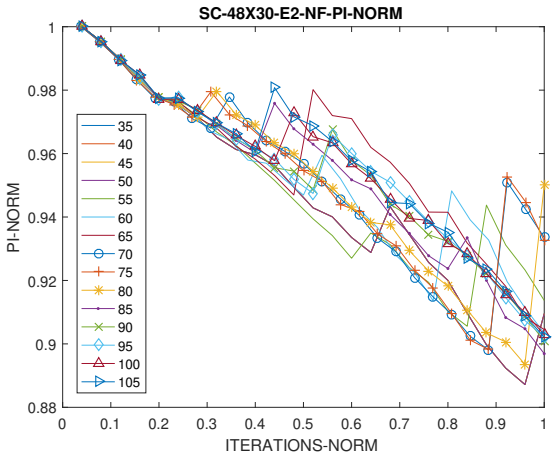
(a) SC-48X30-E1-NF PI



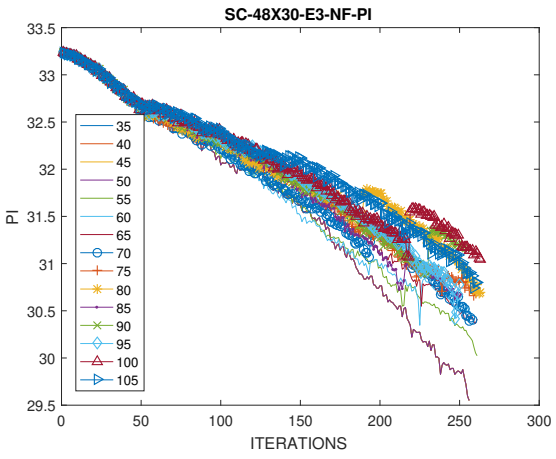
(b) SC-48X30-E1-NF PI normalized



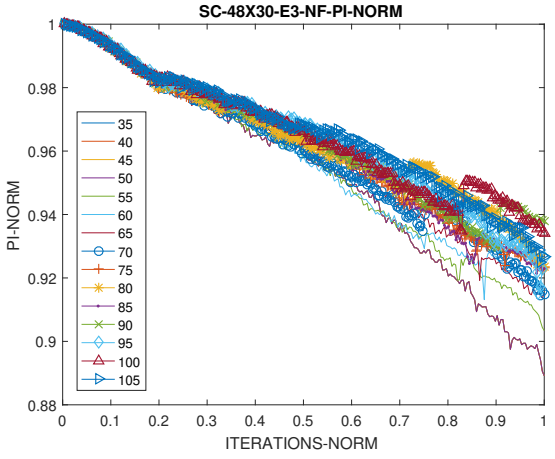
(c) SC-48X30-E2-NF PI



(d) SC-48X30-E2-NF PI normalized

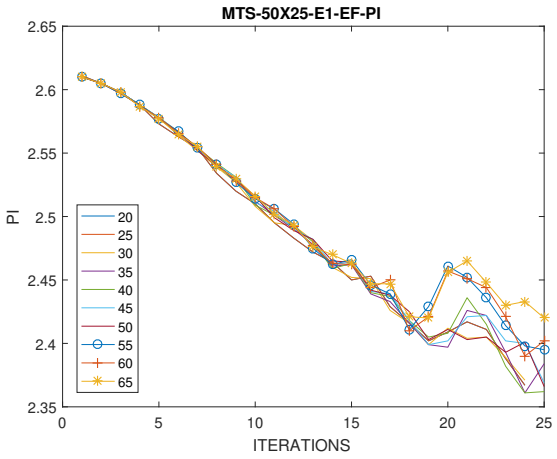


(e) SC-48X30-E3-NF PI

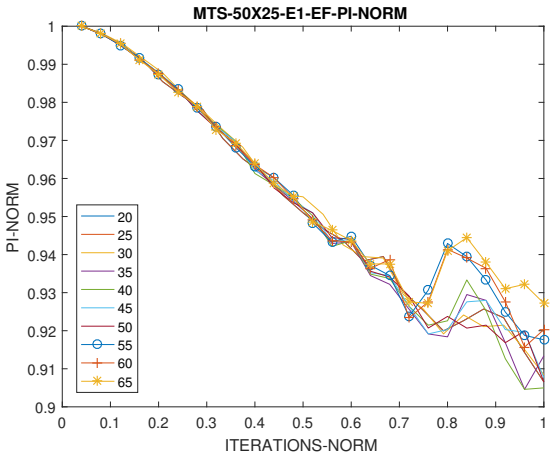


(f) SC-48X30-E3-NF PI normalized

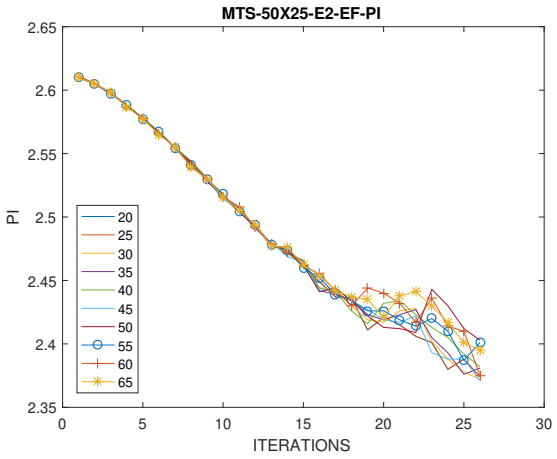
Figure D.4: Nodal filter radius results for SC-48X30



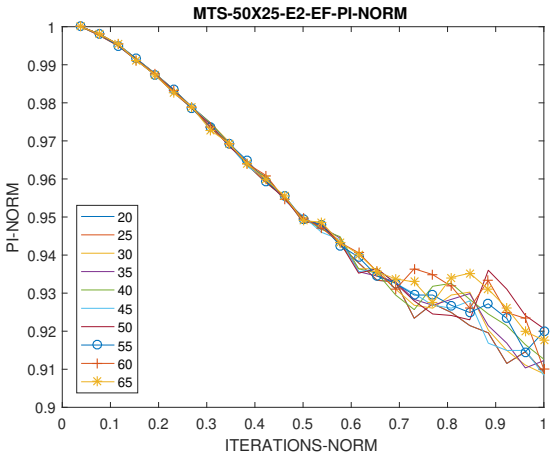
(a) MTS-50X25-E1-EF PI



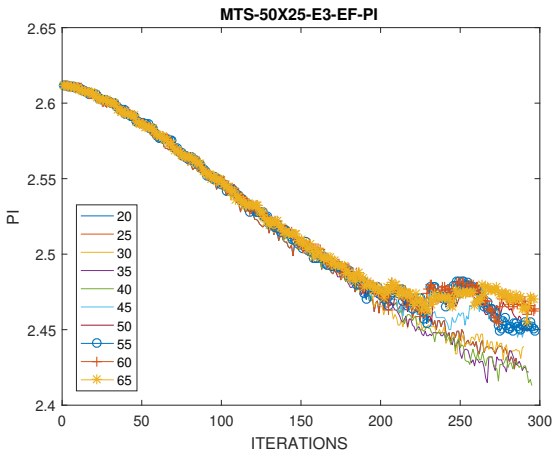
(b) MTS-50X25-E1-EF PI normalized



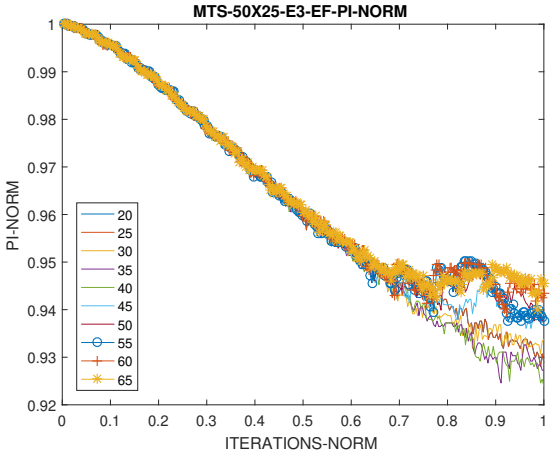
(c) MTS-50X25-E2-EF PI



(d) MTS-50X25-E2-EF PI normalized

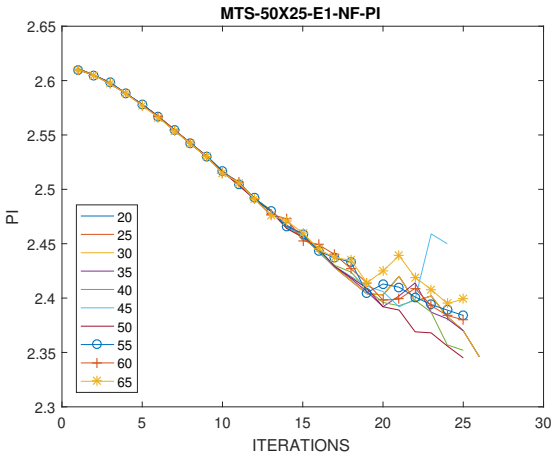


(e) MTS-50X25-E3-EF PI

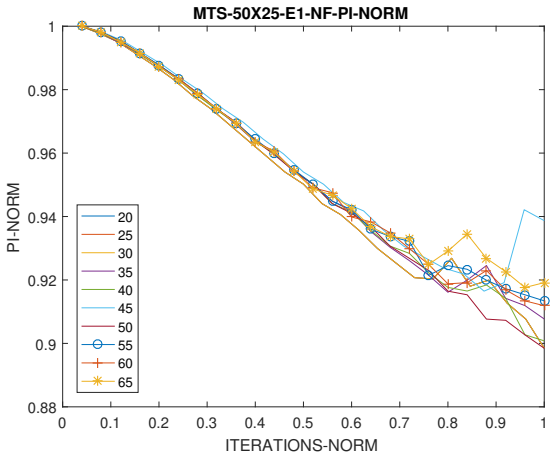


(f) MTS-50X25-E3 PI normalized

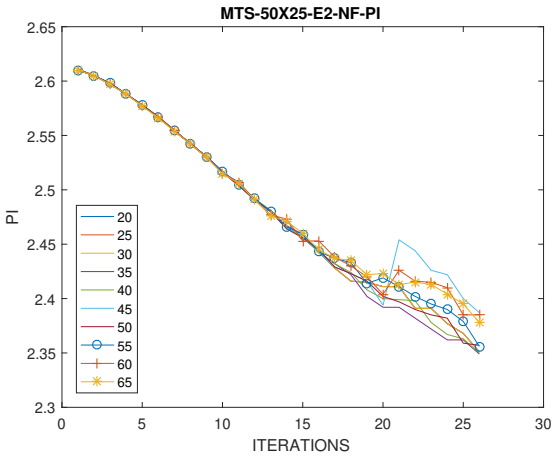
Figure D.5: Elemental filter radius results for MTS-50X25



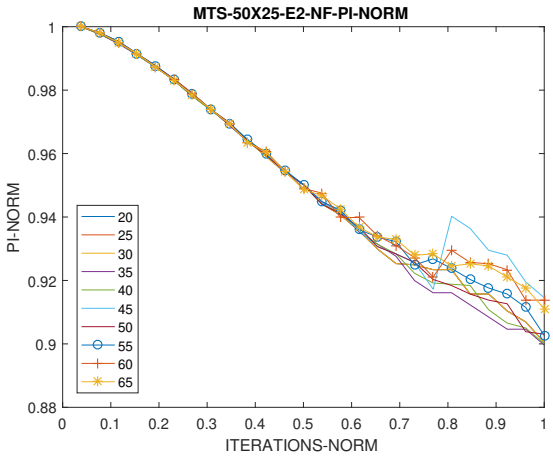
(a) MTS-50X25-E1-NF PI



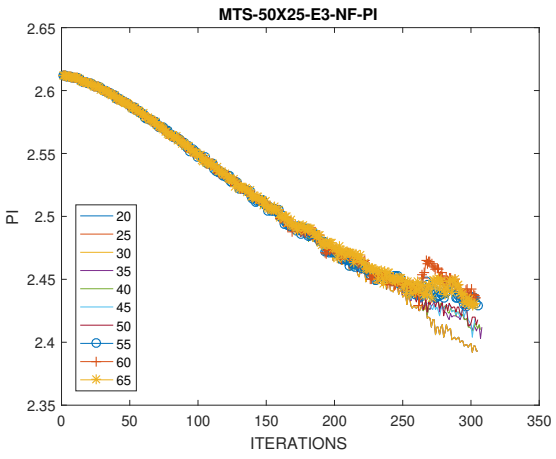
(b) MTS-50X25-E1-NF PI normalized



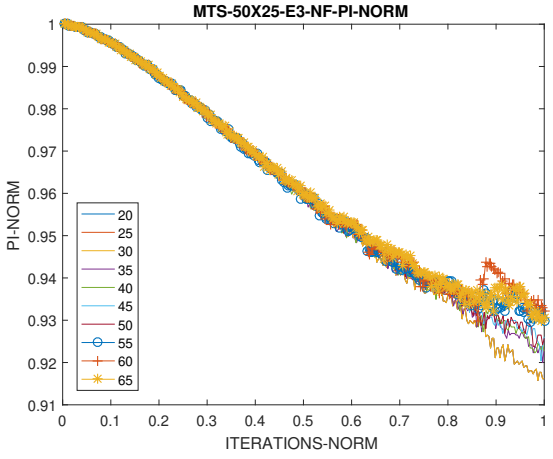
(c) MTS-50X25-E2-NF PI



(d) MTS-50X25-E2-NF PI normalized

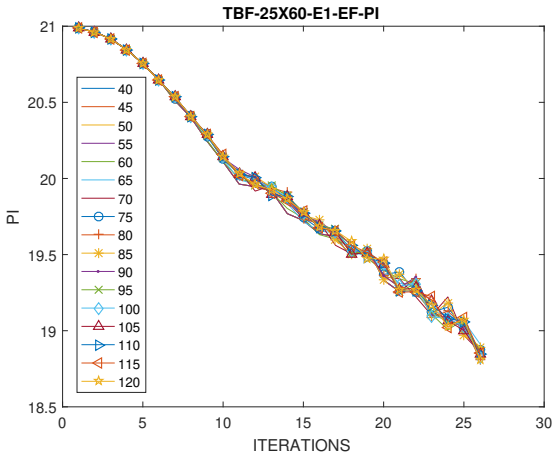


(e) MTS-50X25-E3-NF PI

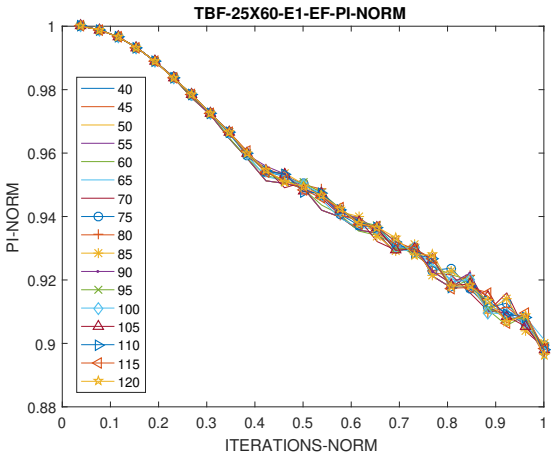


(f) MTS-50X25-E3-NF PI normalized

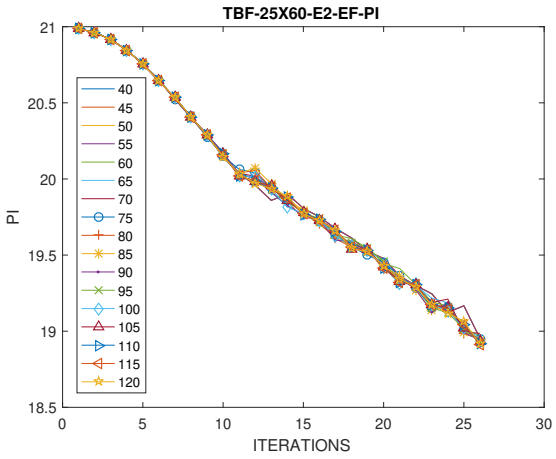
Figure D.6: Nodal filter radius results for MTS-50X25



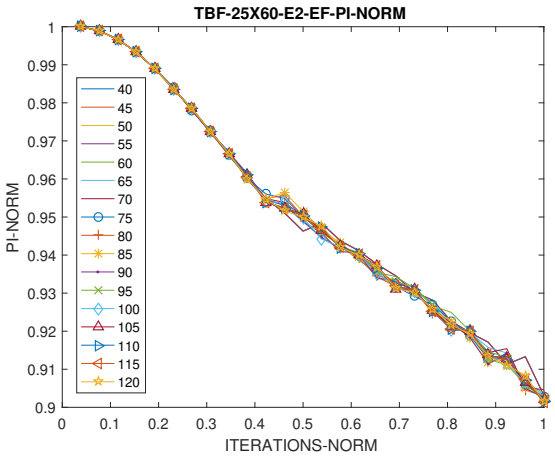
(a) TBF-25X60-E1-EF PI



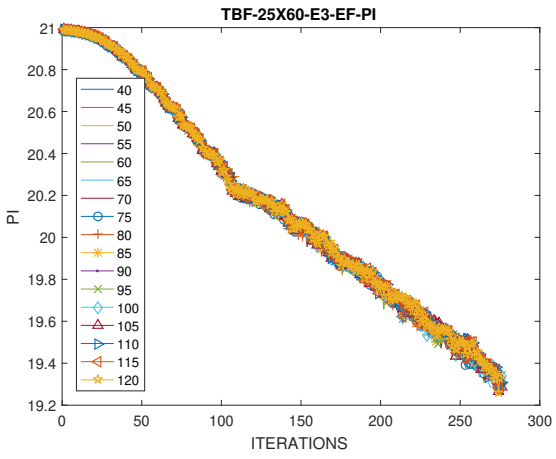
(b) TBF-25X60-E1-EF PI normalized



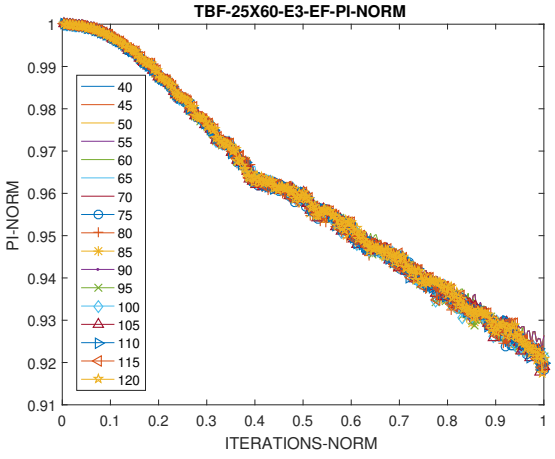
(c) TBF-25X60-E2-EF PI



(d) TBF-25X60-E2-EF PI normalized

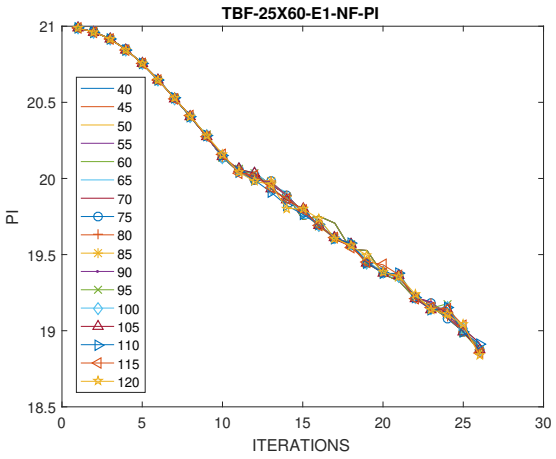


(e) TBF-25X60-E3-EF PI

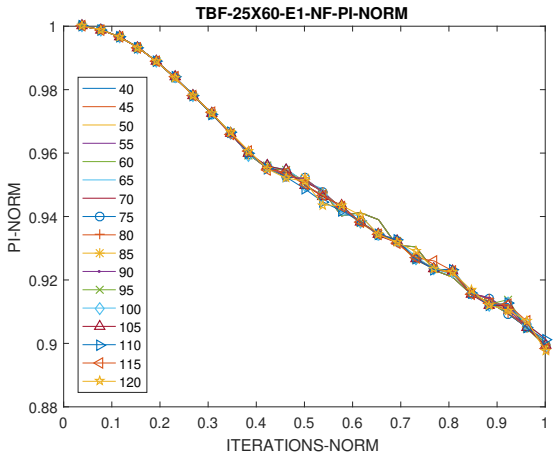


(f) TBF-25X60-E3-EF PI normalized

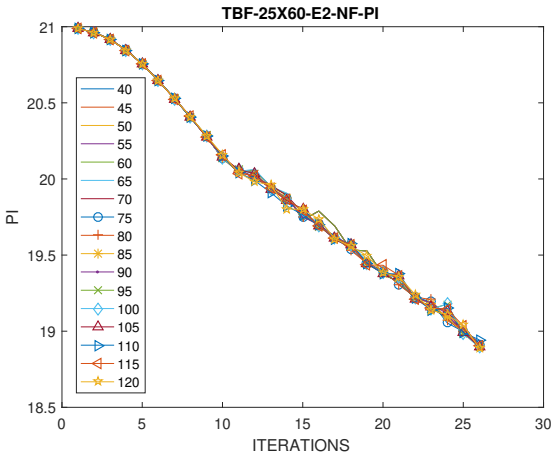
Figure D.7: Elemental filter radius results for TBF-25X60



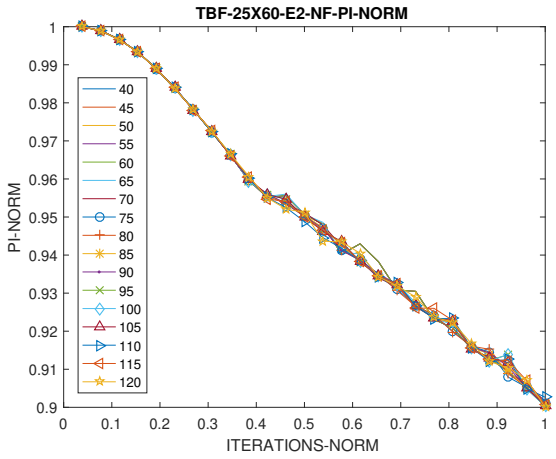
(a) TBF-25X60-E1-NF PI



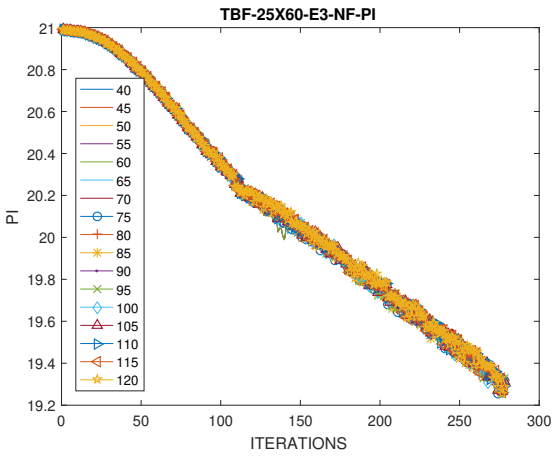
(b) TBF-25X60-E1-NF PI normalized



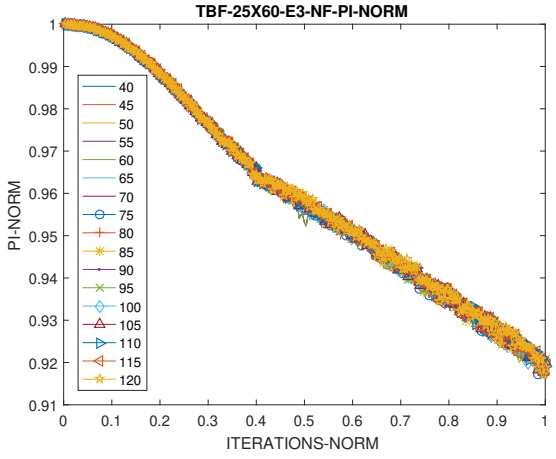
(c) TBF-25X60-E2-NF PI



(d) TBF-25X60-E2-NF PI normalized

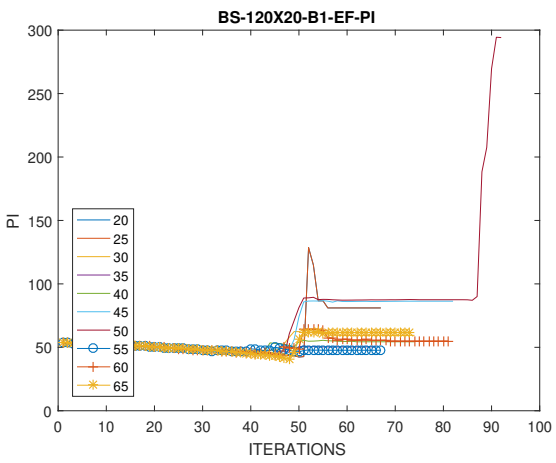


(e) TBF-25X60-E3-NF PI

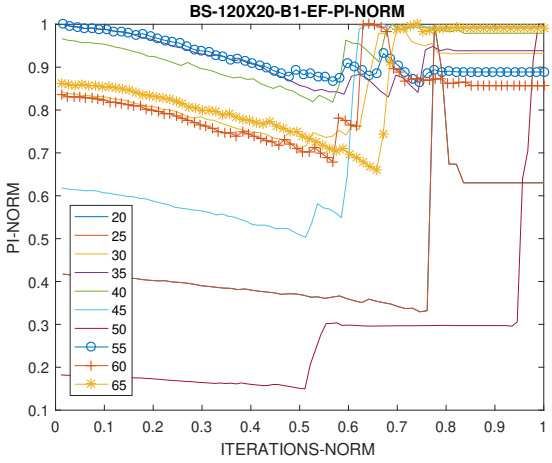


(f) TBF-25X60-E3-NF PI normalized

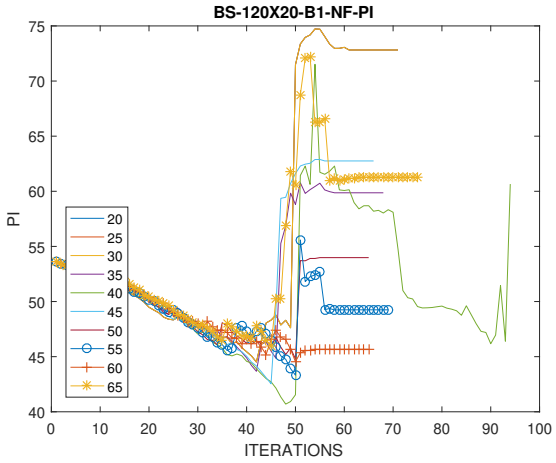
Figure D.8: Nodal filter radius results for TBF-25X60



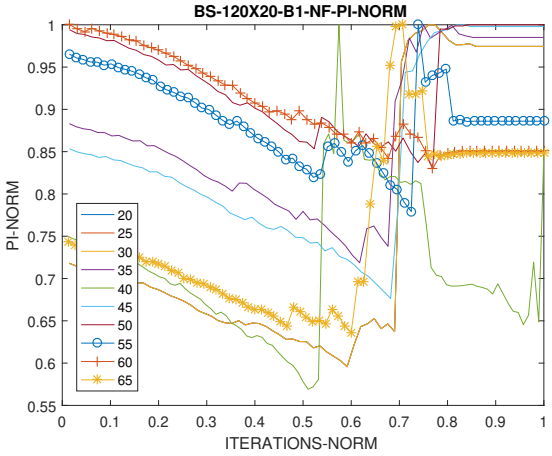
(a) BS-120X20-B1-EF PI



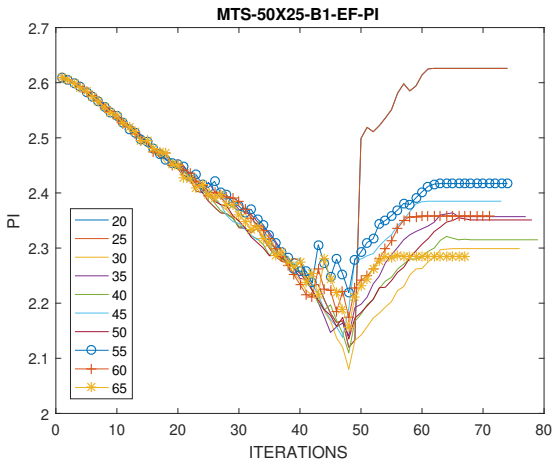
(b) BS-120X20-B1-EF PI normalized



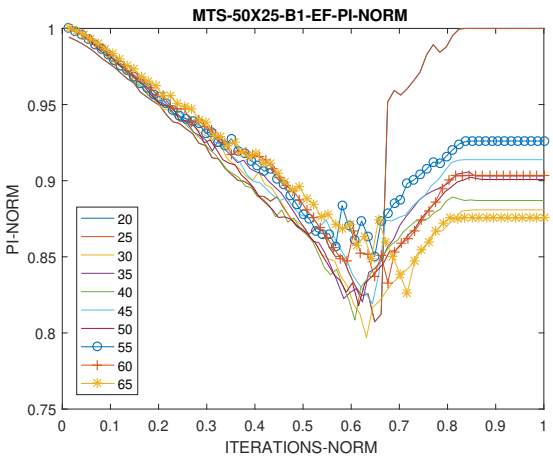
(c) BS-120X20-B12-NF PI



(d) BS-120X20-B1-NF PI normalized

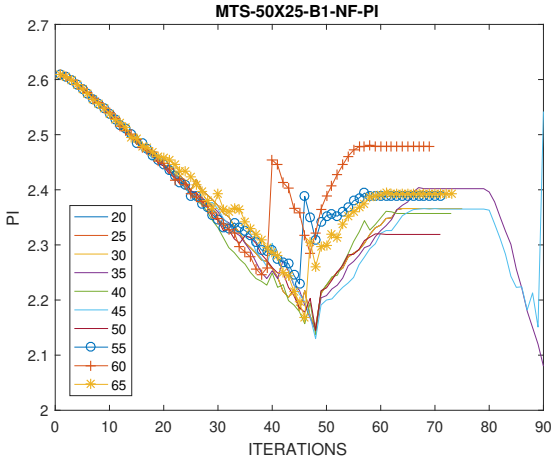


(e) MTS-50X25-B1-EF-EF PI

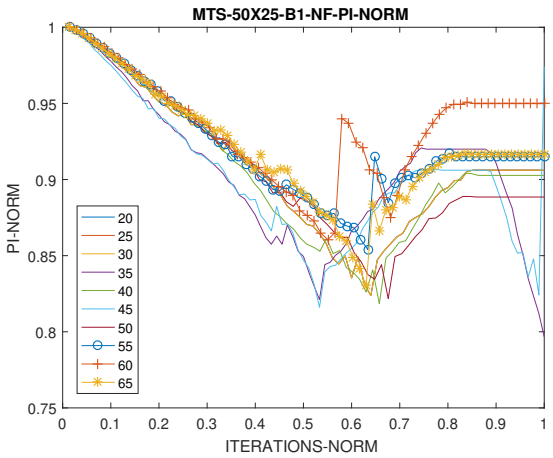


(f) MTS-50X25-B1-EF PI normalized

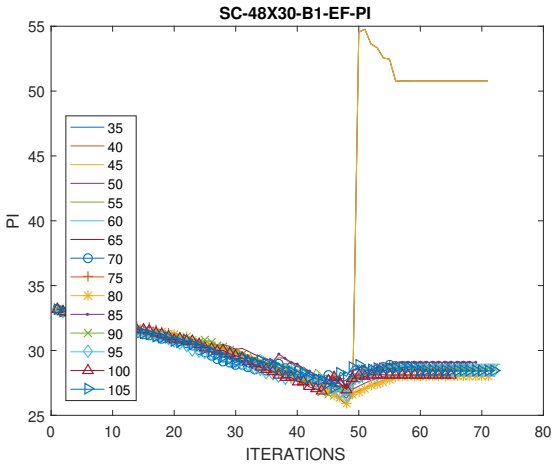
Figure D.9: Filter radius results for BESO BS-120X20 and MTS-50X25



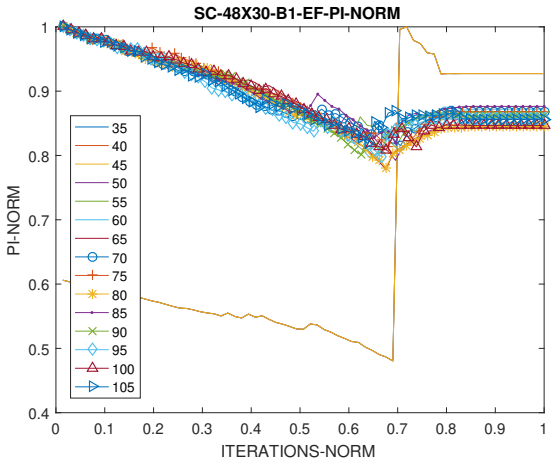
(a) MTS-50X25-B1-NF PI



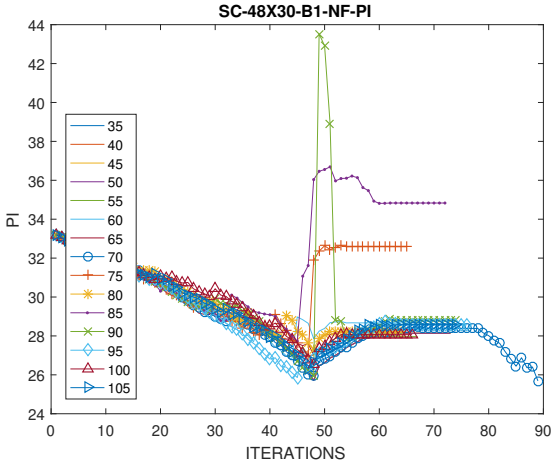
(b) MTS-50X25-B1-NF PI normalized



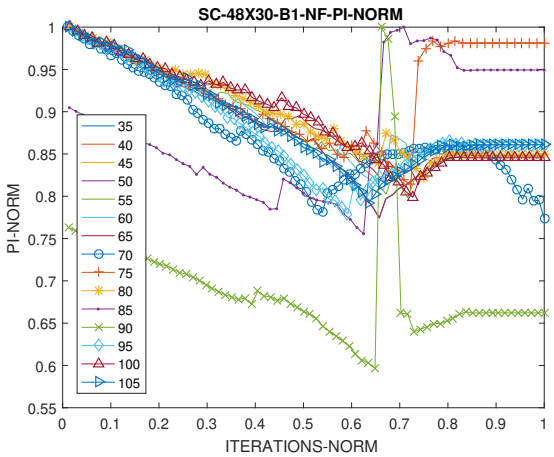
(c) SC-48X30-B1-EF PI



(d) SC-48X30-B1-EF PI normalized

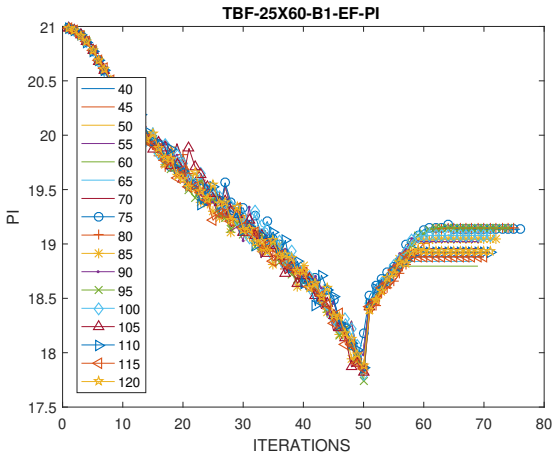


(e) SC-48X30-B1-NF PI

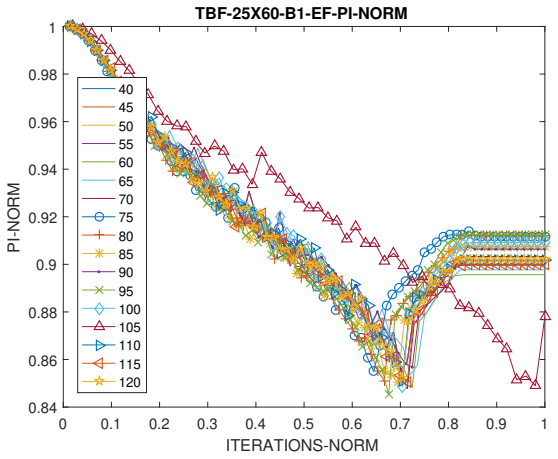


(f) SC-48X30-B1-NF PI normalized

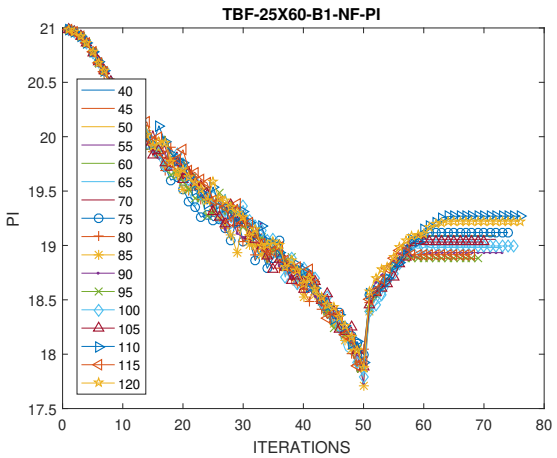
Figure D.10: Filter radius results for BESO MTS-50X25 and SC-48X30



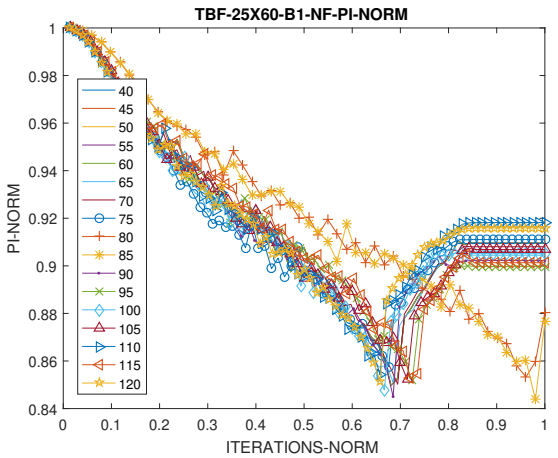
(a) TBF-25X60-B1-NF PI



(b) TBF-25X60-B1-NF PI normalized

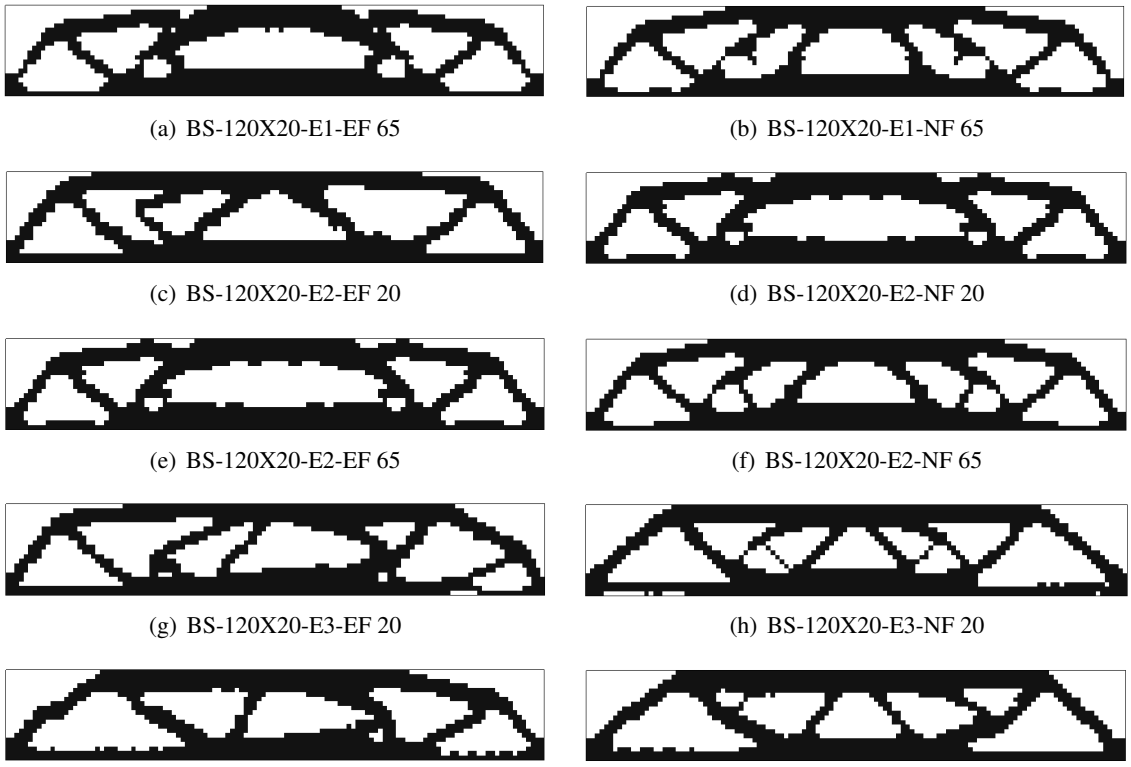


(c) TBF-25X60-B1-EF PI



(d) TBF-25X60-B1-EF PI normalized

Figure D.11: Filter radius results for BESO TBF-25x60



(a) BS-120X20-E1-EF 65

(b) BS-120X20-E1-NF 65

(c) BS-120X20-E2-EF 20

(d) BS-120X20-E2-NF 20

(e) BS-120X20-E2-EF 65

(f) BS-120X20-E2-NF 65

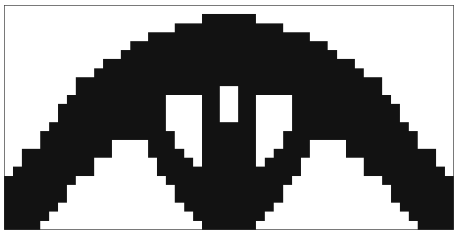
(g) BS-120X20-E3-EF 20

(h) BS-120X20-E3-NF 20

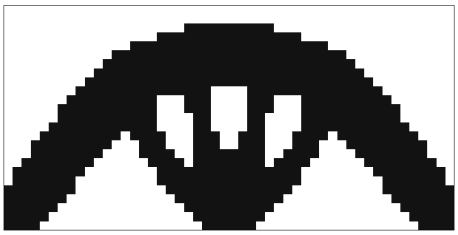
(i) BS-120X20-E3-EF 65

(j) BS-120X20-E3-NF 65

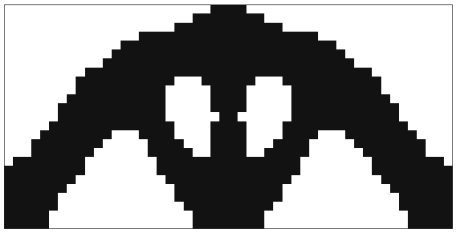
Figure D.12: Filter radius visual results for BS-120X20



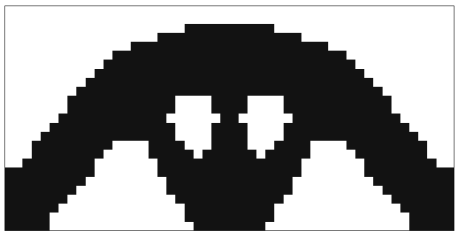
(a) MTS-50X25-E1-EF 20



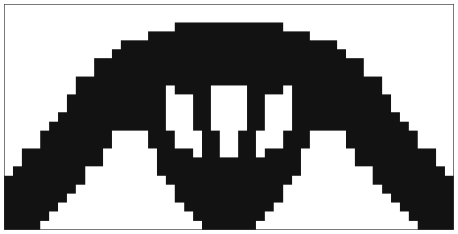
(b) MTS-50X25-E1-NF 20



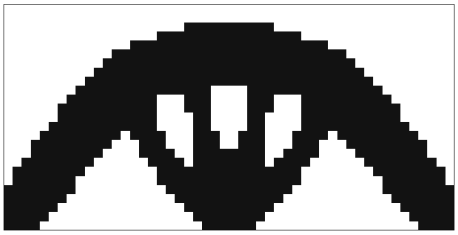
(c) MTS-50X25-E1-EF 65



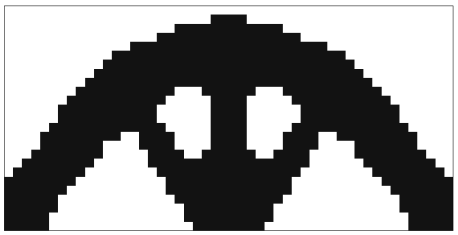
(d) MTS-50X25-E1-NF 65



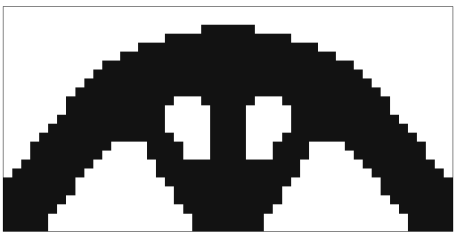
(e) MTS-50X25-E2-EF 20



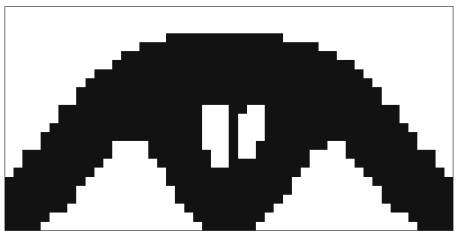
(f) MTS-50X25-E2-NF 20



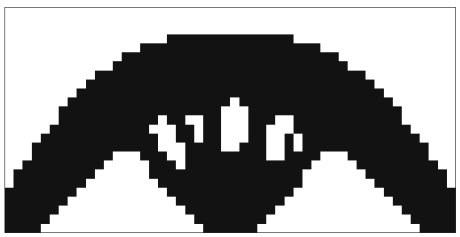
(g) MTS-50X25-E2-EF 65



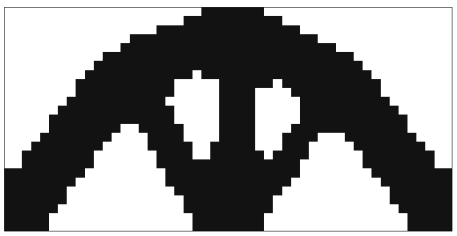
(h) MTS-50X25-E2-NF 65



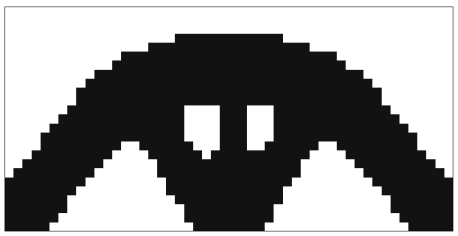
(i) MTS-50X25-E3-EF 20



(j) MTS-50X25-E3-NF 20

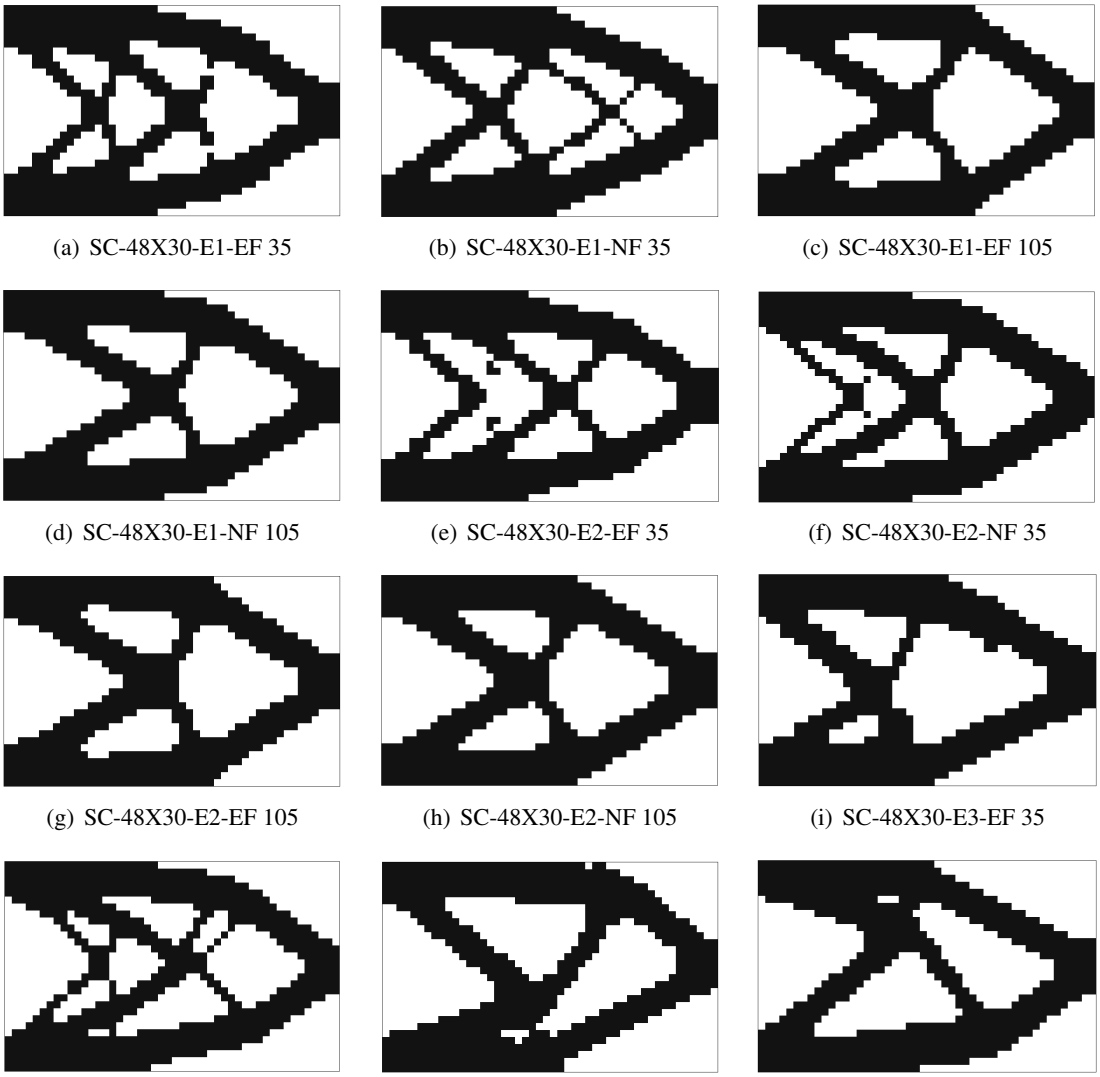


(k) MTS-50X25-E3-EF 65



(l) MTS-50X25-E3-NF 65

Figure D.13: Filter radius visual results for MTS-50X25



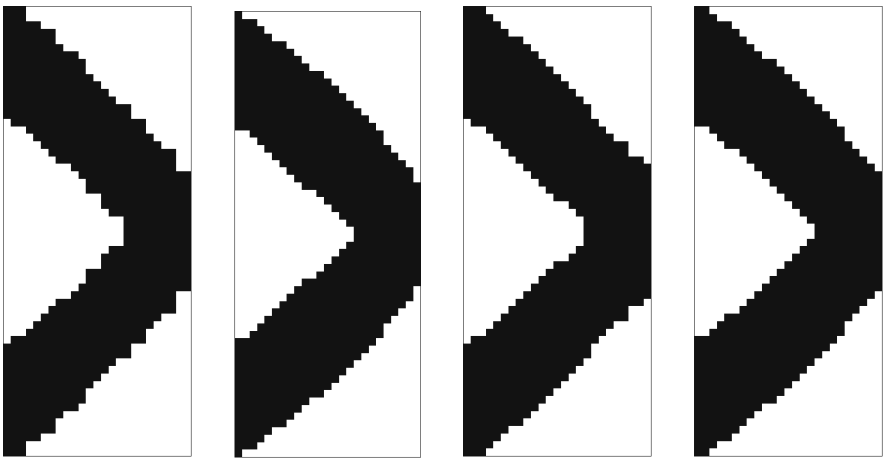
(a) SC-48X30-E1-EF 35 (b) SC-48X30-E1-NF 35 (c) SC-48X30-E1-EF 105

(d) SC-48X30-E1-NF 105 (e) SC-48X30-E2-EF 35 (f) SC-48X30-E2-NF 35

(g) SC-48X30-E2-EF 105 (h) SC-48X30-E2-NF 105 (i) SC-48X30-E3-EF 35

(j) SC-48X30-E3-NF 35 (k) SC-48X30-E3-EF 105 (l) SC-48X30-E3-NF 105

Figure D.14: Filter radius visual results for SC-48X30

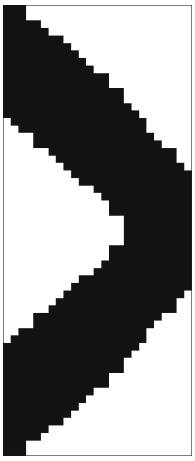


(a) TBF-25X60-E1-EF 40

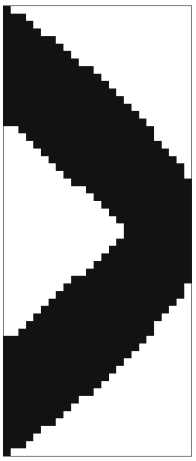
(b) TBF-25X60-E1-NF 40

(c) TBF-25X60-E1-EF 120

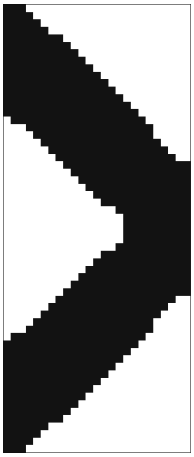
(d) TBF-25X60-E1-NF 120



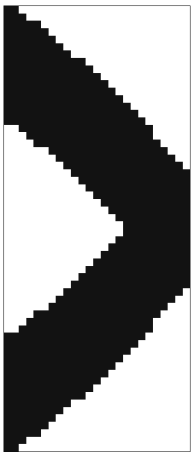
(e) TBF-25X60-E2-EF 40



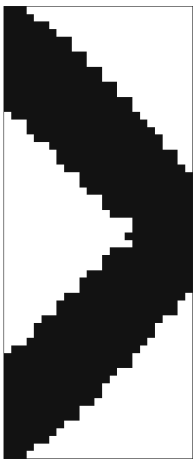
(f) TBF-25X60-E2-NF 40



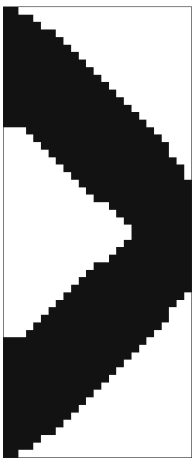
(g) TBF-25X60-E2-EF 120



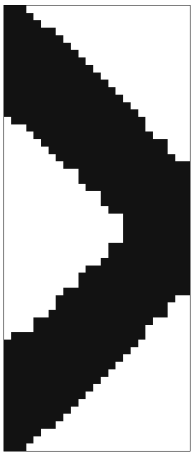
(h) TBF-25X60-E2-NF 120



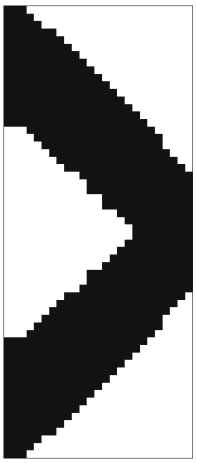
(i) TBF-25X60-E3-EF 40



(j) TBF-25X60-E3-NF 40



(k) TBF-25X60-E3-EF 120

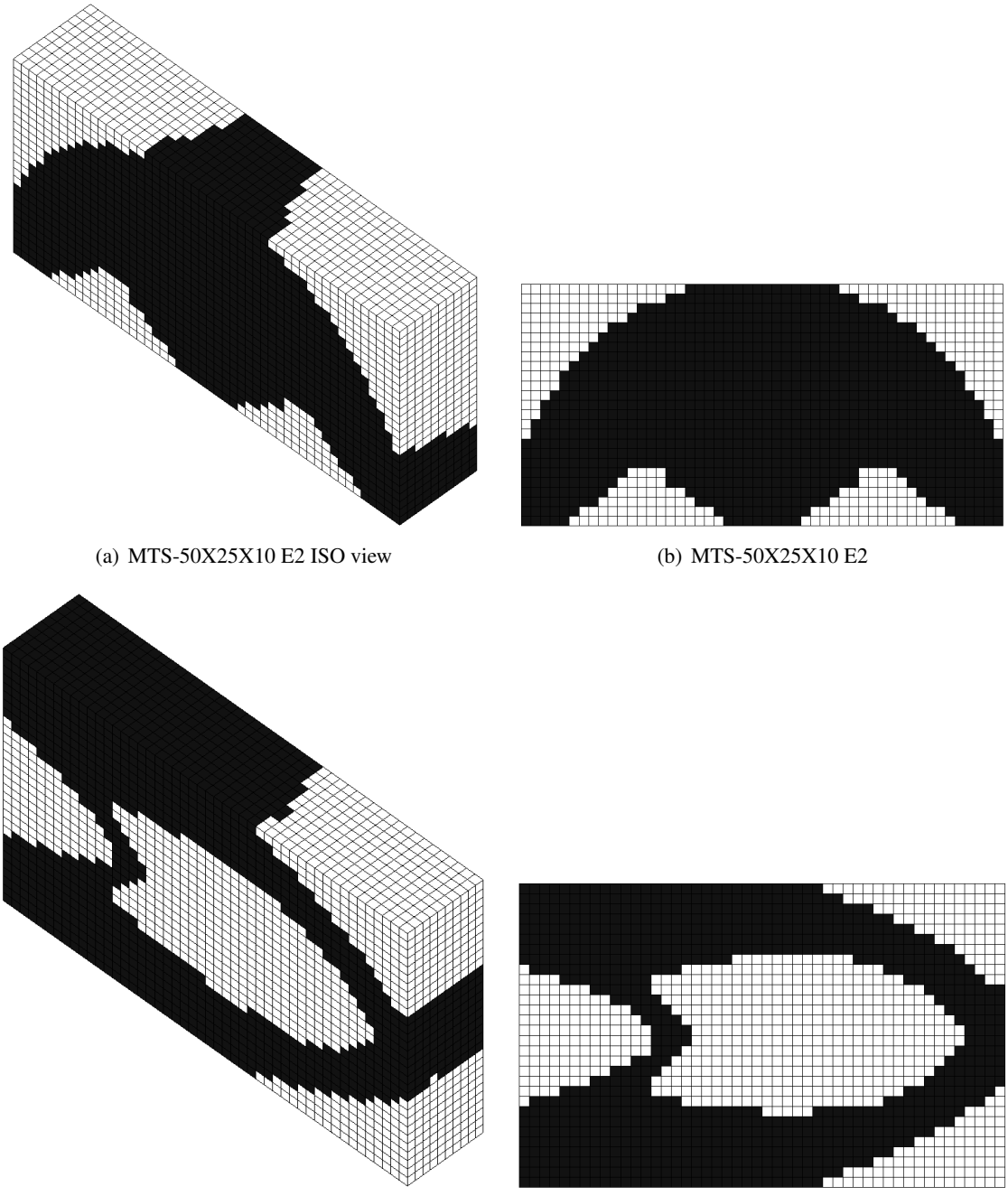


(l) TBF-25X60-E3-NF 120

Figure D.15: Filter radius visual results for TBF-25X60

Appendix E

3D Models



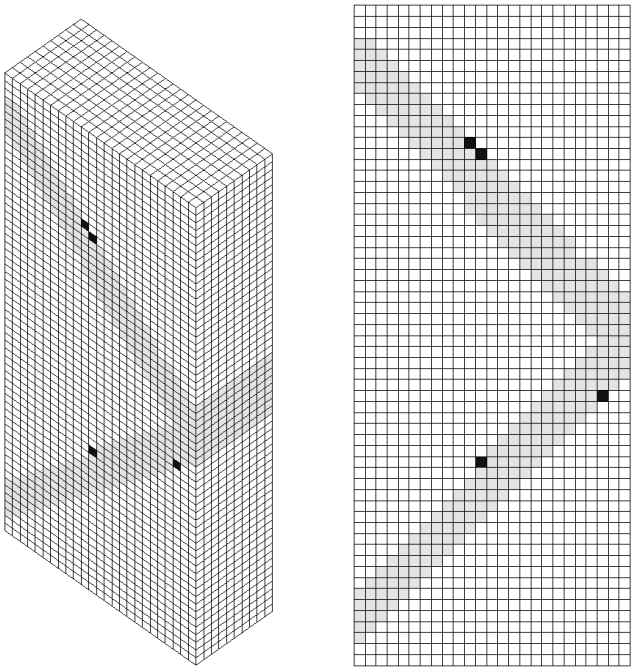
(a) MTS-50X25X10 E2 ISO view

(b) MTS-50X25X10 E2

(c) SC-48X30X10 E2

(d) SC-48X30X10 E2

Figure E.1: 3D final design for MTS-50X25X10



(a) TBF-25X60X10 B1 ISO view

(b) TBF-25X60X10 B1

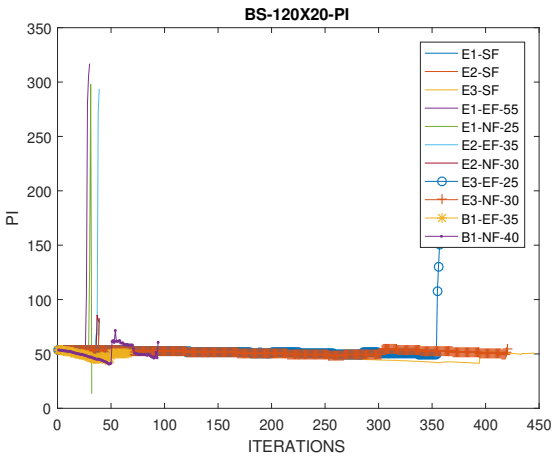
Figure E.2: 3D final design for TBF-25X60X10

Appendix F

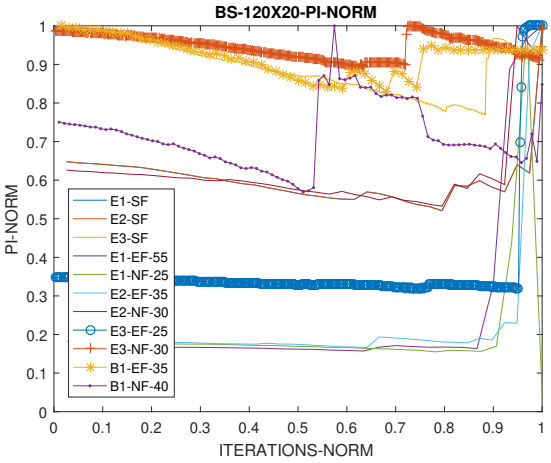
Final comparison

Table F.1: Final comparison cases abbreviation

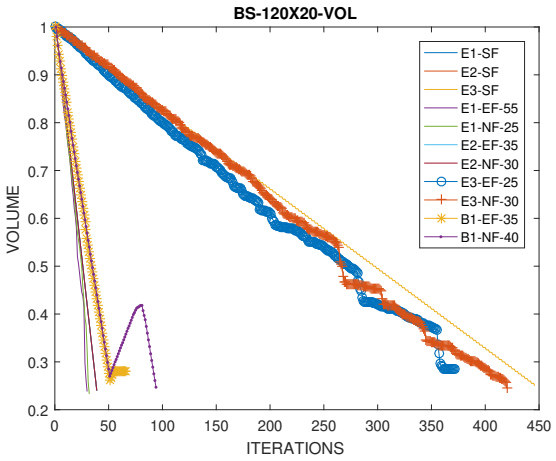
BS-120X20	MTS-50X25	SC-48X30	TBF
BS-120X20-E1-SF	MTS-50X25-E1-SF	SC-48X30-E1-SF	TBF-25X60-E1-SF
BS-120X20-E2-SF	MTS-50X25-E2-SF	SC-48X30-E2-SF	TBF-25X60-E2-SF
BS-120X20-E3-SF	MTS-50X25-E3-SF	SC-48X30-E3-SF	TBF-25X60-E3-SF
BS-120X20-E1-EF-55	MTS-50X25-E1-EF-35	SC-48X30-E1-EF-50	TBF-25X60-E1-EF-55
BS-120X20-E1-NF-25	MTS-50X25-E1-NF-30	SC-48X30-E1-NF-50	TBF-25X60-E1-NF-55
BS-120X20-E2-EF-35	MTS-50X25-E2-EF-25	SC-48X30-E2-EF-70	TBF-25X60-E2-EF-55
BS-120X20-E2-NF-30	MTS-50X25-E2-NF-35	SC-48X30-E2-NF-55	TBF-25X60-E2-NF-55
BS-120X20-E3-EF-25	MTS-50X25-E3-EF-35	SC-48X30-E3-EF-75	TBF-25X60-E3-EF-55
BS-120X20-E3-NF-30	MTS-50X25-E3-NF-30	SC-48X30-E3-NF-50	TBF-25X60-E3-NF-55
BS-120X20-B1-EF-35	MTS-50X25-B1-EF-30	SC-48X30-B1-EF-45	TBF-25X60-B1-EF-95
BS-120X20-B1-NF-40	MTS-50X25-B1-NF-45	SC-48X30-B1-NF-90	TBF-25X60-B1-NF-90



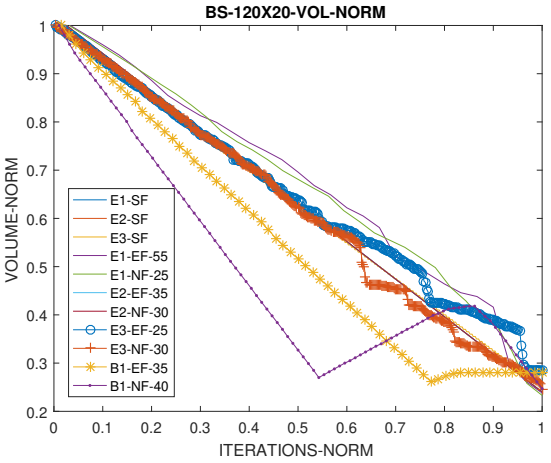
(a) BS-120X20 PI



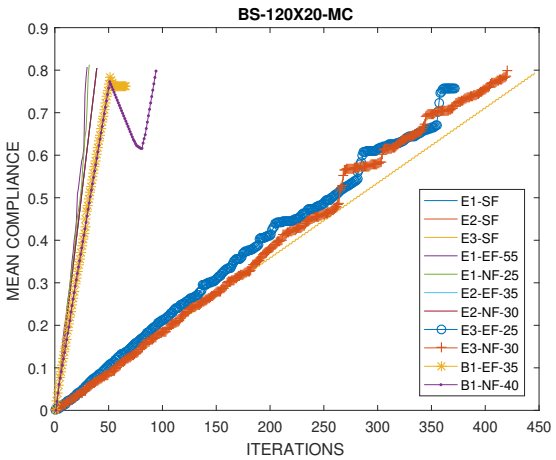
(b) BS-120X20 PI normalized



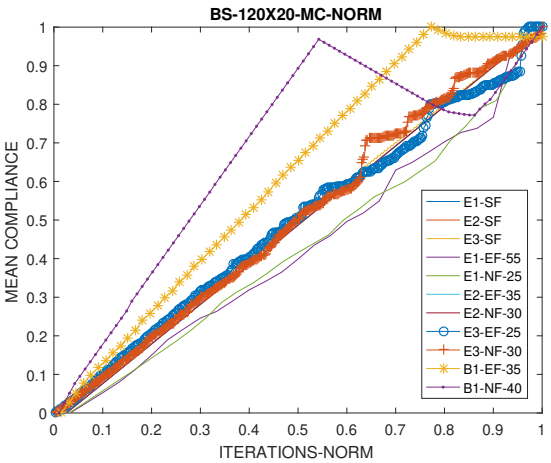
(c) BS-120X20 Volume



(d) BS-120X20 Volume normalized

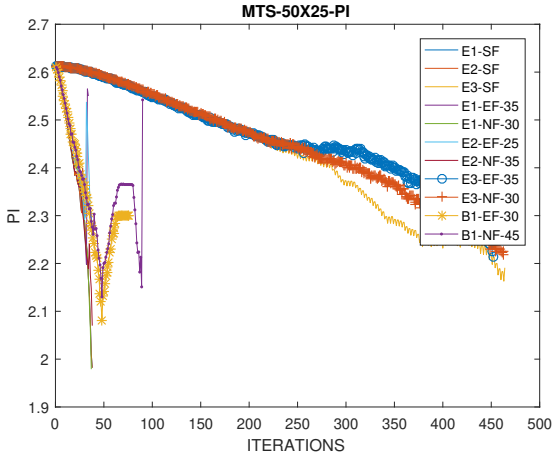


(e) BS-120X20 Mean compliance

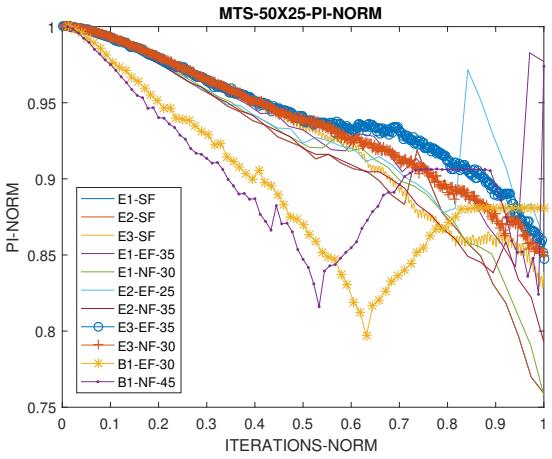


(f) BS-120X20 Mean compliance normalized

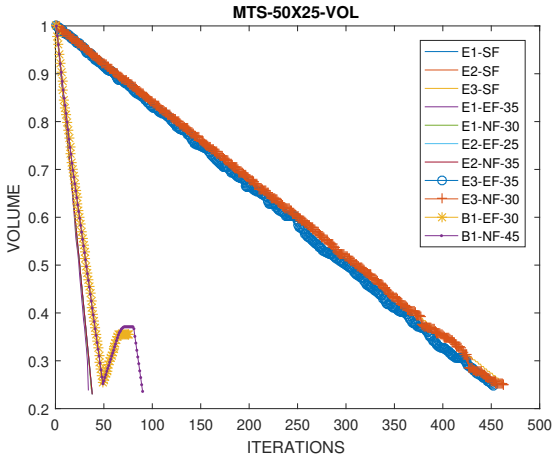
Figure F.1: Final comparison for BS-120X20



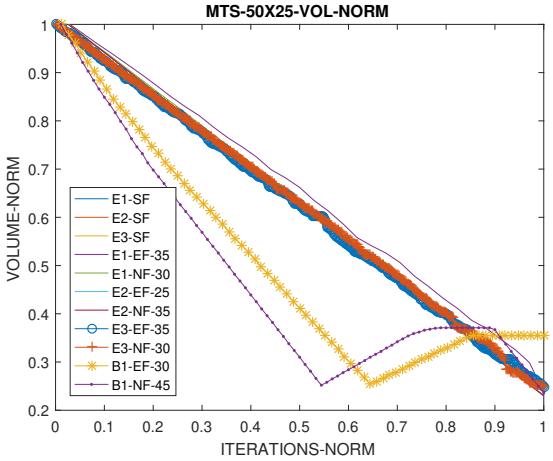
(a) MTS-50X25 PI



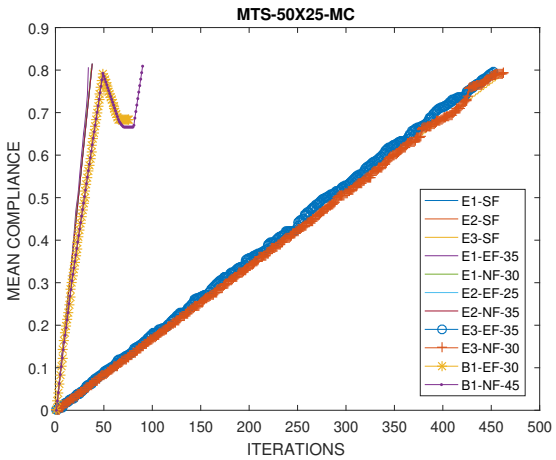
(b) MTS-50X25 PI normalized



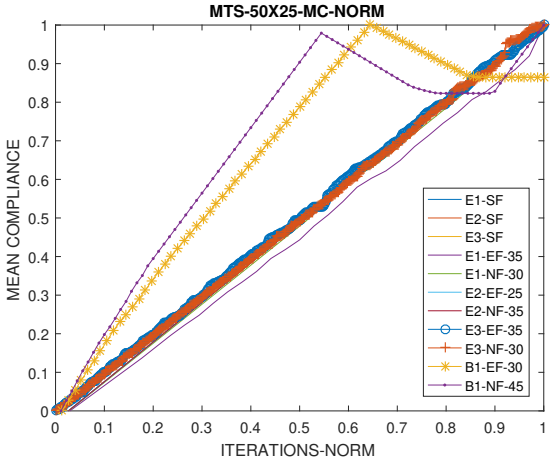
(c) MTS-50X25 Volume



(d) MTS-50X25 Volume normalized

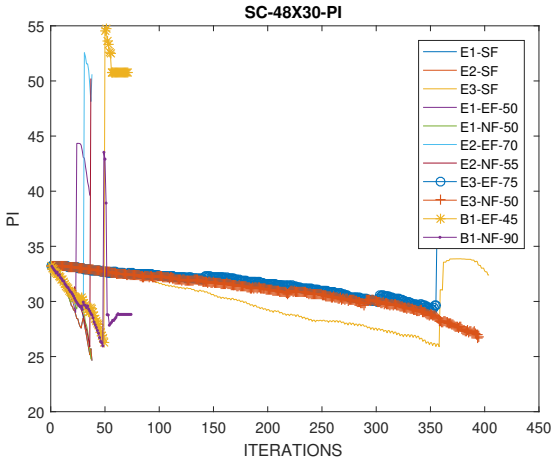


(e) MTS-50X25 Mean compliance

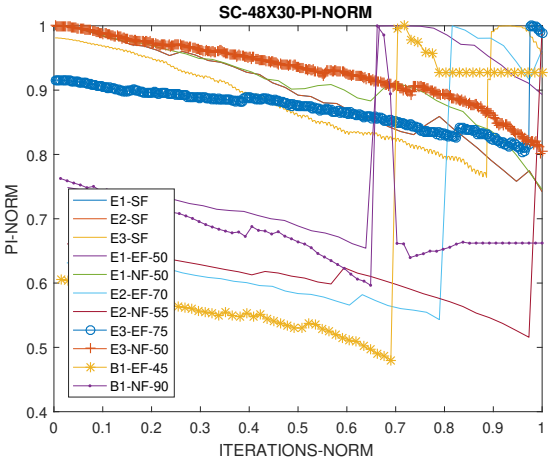


(f) MTS-50X25 Mean compliance normalized

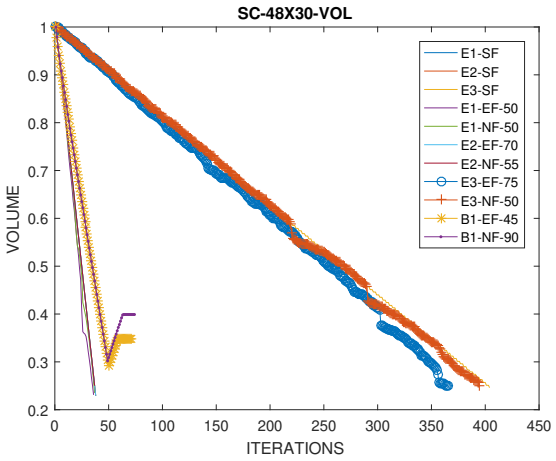
Figure F.2: Final comparison for MTS-50X20



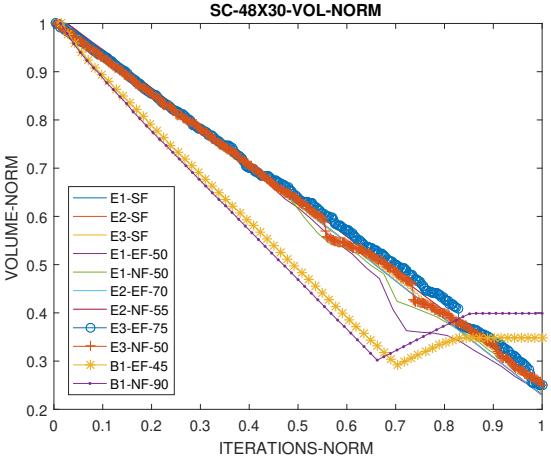
(a) SC-48X30 PI



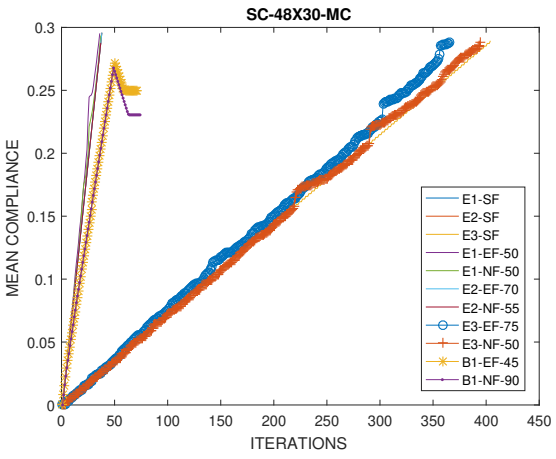
(b) SC-48X30 PI normalized



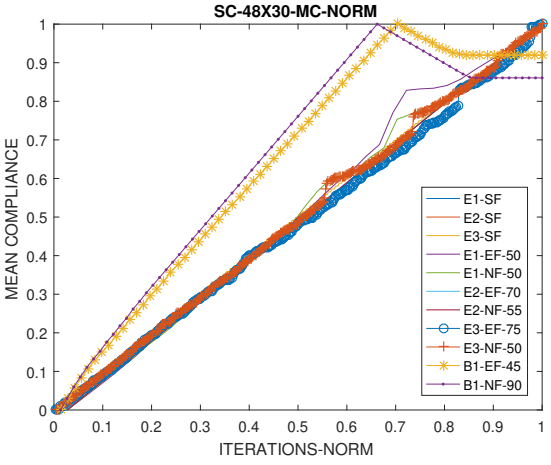
(c) SC-48X30 Volume



(d) SC-48X30 Volume normalized

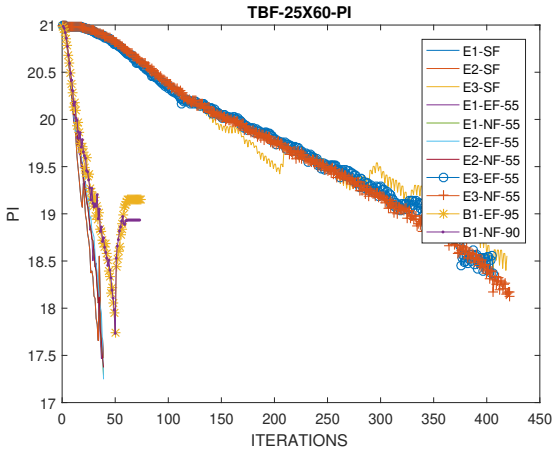


(e) SC-48X30 Mean compliance

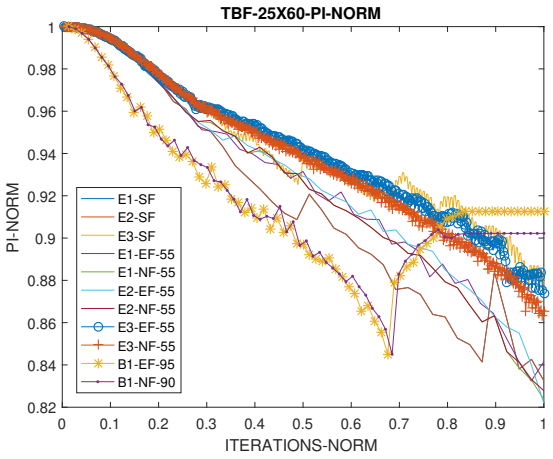


(f) SC-48X30 Mean compliance normalized

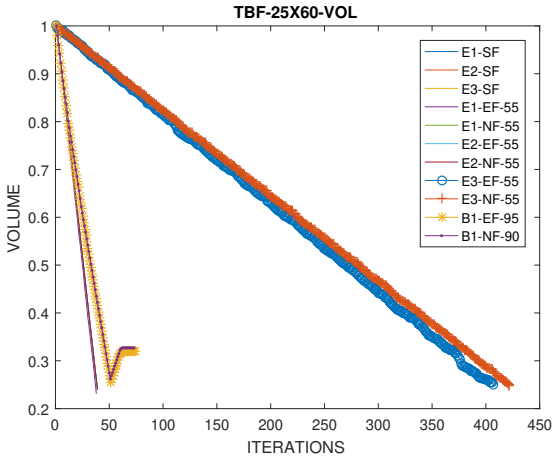
Figure F.3: Final comparison for SC-48X30



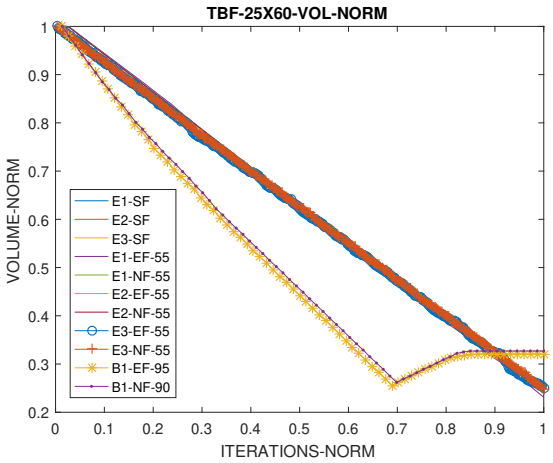
(a) TBF-25X60 PI



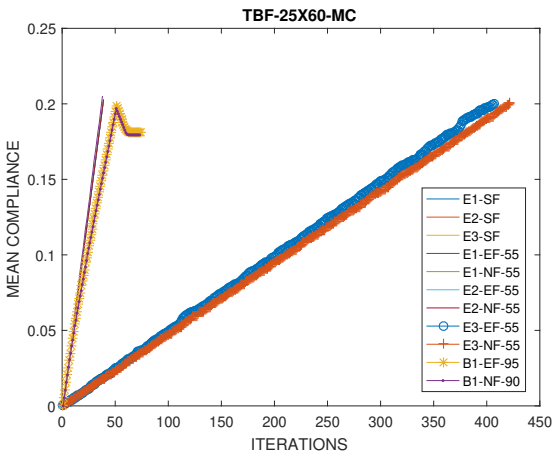
(b) TBF-25X60 PI normalized



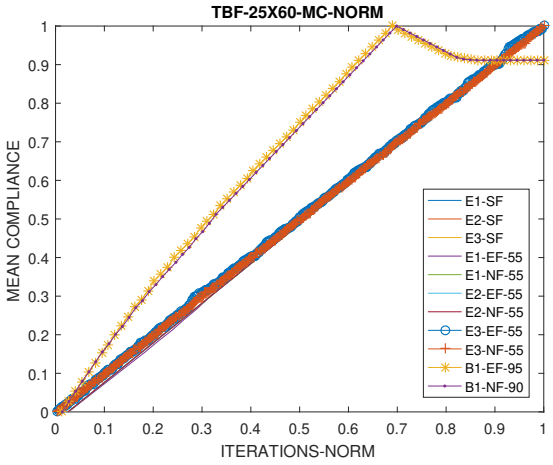
(c) TBF-25X60 Volume



(d) TBF-25X60 Volume normalized



(e) TBF-25X60 Mean compliance



(f) TBF-25X60 Mean compliance normalized

Figure F.4: Final comparison for TBF-25X60

Bibliography

V. Beiranvand, W. Hare, and Y. Lucet. Best practices for comparing optimization algorithms. *Optim. Eng.*, 18(4):815–848, 2017.

M. P. Bendsøe and O. Sigmund. *Topology Optimization: Theory, Method and Application*. Springer, 2003.

P. Christensen and K. Anders. *An Introduction to Structural Optimization*, volume 153 of *Solid Mechanics and Its Applications*. Springer Netherlands, 2008.

D. Chu, Y. Xie, A. Hira, and G. Steven. Evolutionary structural optimization for problems with stiffness constraints. *Finite Elements in Analysis and Design*, 21(4):239 – 251, 1996.

R. W. Clough and E. L. Wilson. Early Finite Element Research at Berkeley. *Fifth U.S. Natl. Conf. Comput. Mech.*, pages 1–35, 2000.

DS SIMULIA Corp. Abaqus 6.11. Theory manual. *ABAQUS, Inc. Dassault Systèmes*, page 1172, 2011.

J. Fish and T. Belytschko. *A First Course in Finite Elements*. John Wiley & Sons, Ltd, 2007.

J. E. Gordon and M. B. Stewart. *Structures or Why Things Don ' t Fall Down*, volume 48. Pelican Books, 1980.

R. B. Haber, C. S. Jog, and M. P. Bendsoe. A new approach to variable-topology shape design using a constraint on perimeter. *Struct. Optim.*, 11:1–12, 1996.

X. Huang and Y. Xie. *Evolutionary Topology Optimisation of Continuum Structures. Methods and Applications*. Wiley, 2010a.

X. Huang and Y. M. Xie. Evolutionary topology optimization of continuum structures with an additional displacement constraint. *Struct. Multidiscip. Optim.*, 40, 2010b.

C. S. Jog. Topology design of structures using a dual algorithm and a constraint on the perimeter. *Int. J. Numer. Methods Eng.*, 54(7):1007–1019, 2002.

T. Johns. JOURNAL OF THE AERONAUTICAL SCIENCES Turbulence in Supersonic Flow *. *J. Aeronaut. Sci.*, 23(9):657–674,682, 1953.

U. Kirsch. *Structural optimization: fundamentals and applications*, volume 53. 1993.

S. R. Labanda and M. Stolpe. Benchmarking of optimization methods for topology optimization problems. *11th World Congr. Comput. Mech.*, page 23, 2014.

Q. Li, G. P. Steven, and Y. M. Xie. A simple checkerboard suppression algorithm for evolutionary structural optimization. *Struct. Multidiscip. Optim.*, 22(3):230–239, 2001.

A. Mirmiran, Y. Shao, M. Shahawy, J.-S. Kim, H.-J. Yoon, and K.-B. Shin. ABAQUS Theory manual. *ABAQUS, Inc. Dassault Systèmes*, page 841, 2007.

O. M. Querin, V. Young, G. P. Steven, and Y. M. Xie. Computational efficiency and validation of bi-directional evolutionary structural optimization. *Comput. Methods Appl. Mech. Eng.*, 189 (2):559–573, 2000.

S. Rojas-Labanda and M. Stolpe. Benchmarking optimization solvers for structural topology optimization. *Struct. Multidiscip. Optim.*, 52(3):527–547, 2015.

Rozvany and G.I.N. A critical review of established methods of structural topology optimization. *Struct. Multidiscip. Optim.*, 37(3):217–237, 2009.

O. Sigmund. On the design of compliant mechanisms using topology optimization. *Mechanics of Structures and Machines*, 25(4):493–524, 1997. doi: 10.1080/08905459708945415.

O. Sigmund and J. Petersson. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Struct. Optim.*, 16(1):68–75, 1998.

Simulia. Abaqus 6.14 Online Documentation, 2014. URL `https://www.sharcnet.ca/Software/Abaqus/6.14.2/v6.14/`.

S. Timoshenko and J. Goodier. *Theory of elasticity*. McGraw-Hill, 1971.

Y. M. Xiet and G. P. Steven. A Simple Approach To Structural Optimization. *Comput. Struct.*, 49 (5):885–896, 1992.

X. Y. Yang, O. M. Querin, G. P. Steven, and Y. M. Xei. Bidirectional Evolutionary Method for Stiffness Optimization. *AIAA Journal*, 37:1483–1488, Nov. 1999.

X. Y. Yang, Y. M. Xie, J. S. Liu, G. T. Parks, and P. J. Clarkson. Perimeter control in the bidirectional evolutionary optimization method. *Struct. Multidiscip. Optim.*, 24(6):430–440, 2003.